

Appl. No. 10/009,397  
Reply to Office Action of January 26, 2007

**APPENDIX**  
**CERTIFIED TRANSLATION OF THE PRIORITY DOCUMENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

TRANSLATOR'S DECLARATION AND CERTIFICATE

APPLICANT: Hofstetter

SERIAL NO.: 10/009,397

FILED: February 25, 2002

TITLE: "NETWORK, INTERPRETER FOR SUCH A NETWORK  
AND METHOD FOR OPERATING A NETWORK"

Commissioner for Patents  
Box 1450  
Alexandria, VA 22313-1450

SIR:

I, Charles Bullock, declare and state that I am knowledgeable in German and English, and I hereby certify that the attached translation of the attached German Priority Application 199 22 118.9, filed in the German Patent and Trademark Office on 12 May 1999, is truthful and accurate to the best of my knowledge.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

DATE: 26 April 2007



**Schaumburg et al.**  
**Certified translation of priority document**  
**Case No. P01,0402 (26970-0156)**  
**Client Ref. No. 990503 PUS**  
**Inventor: Hofstetter**

Translation / April 26, 2007 / C. Bullock

FEDERAL REPUBLIC OF GERMANY

Priority Document concerning the Submission  
of a Patent Application

File number: 199 22 118.9

Application date: 12 May 1999

Applicant/patent holder: Océ Printing Systems GmbH,  
Poing/DE

Title: Network, interpreter for such a network and  
method for operation of a network

IPC: G 06 F 15/173

The attached pieces are a correct and precise reproduction of the original documents of this application.

München, the 2nd of June 2000  
German Patent and Trademark Office  
The President  
by order  
[signature]  
Dzierzon

**NETWORK, INTERPRETER FOR SUCH A NETWORK AND METHOD****5 FOR OPERATING A NETWORK**

The invention concerns a network, an interpreter for such a network and a method for operating a network. In particular, the invention concerns a network for a printing system.

10

As a rule, a plurality of printers and printing systems and potentially corresponding pre-processing and post-processing devices that, for example, trim the printed paper to a specific format or bind it are connected to the existing data networks such as, for example, LANs (Local Area Networks) and WANs (Wide Area

15 Networks). The individual printers and printing systems differ greatly in terms of their performance capability; for example, ink jet printers with a print output of 4 pages per minute or high-performance printers with a print output of 40 pages or more per minute can thus be connected to the data network. The printers and printing systems differ not only in terms of their printing output but also in terms  
20 of their printing quality. For example, there are thus printers that print [sic] only in a single color (monochrome), in contrast whereunto color printer are also increasingly utilized. Electrophotographic high-performance printer devices are currently in the position of printing two colors without further ado, i.e. in what is referred to as a spot color mode or highlight color mode. Such a printer, for  
25 example, is known by the name PAGESTREAM® 200DSC from Océ Printing Systems GmbH.

The different printers and printing systems are usually also connected to the respective network with different system interfaces such as, for example, SNMP  
30 (Simple Network Management Protocol) or DMI (Desktop Management Interface). Although a physical connection to a plurality of printers or,

respectively, printing systems often exists via the network, only a portion of the printers can be addressed.

With the rapid development of the intranets and internets (with which multiple  
5 LANs and WANs are connected into a single network), particularly due to the introduction of the WWW service based on the HTTP protocol (**Hypertext Transport Protocol**), the number of printers and printing systems that can be addressed in principle is increasing explosively, whereby the multiplicity of protocols for addressing the printers and printing systems increases accordingly.

10

This success of the Internet and of the intranets was greatly promoted by the introduction of the World Wide Web (WWW) that can be simply operated by Internet users and allows the most varied types of information to be fetched from the greatest variety of different locations in the entire world. The WWW service of  
15 the Internet is based on the client-server principle. The communication ensues between a web server (which is also designated as a WWW server and which provides information) and a client that displays the information. The information is stored on the web server in pages, whereby the data in the pages are stored, for example, in what is referred to as HTML format (**Hypertext Markup Language**).

20

Other formats are also in use such as, for example, XML. These formats are derivatives of the underlying SGML format (**Standard General Markup Language**). These formats are designated as markup languages since the presentation can be described and defined with them.

25

A corresponding document is composed of normal text in which control instructions (what are known as "tags") are inserted into the text. Among other things, these tags influence the layout that is displayed later in the viewing program (the browser) at the client. For example, there are thus tags to generate headings or tags that can modify the type face. The tags are always enclosed in "<  
30 ... >".

The transmission of the information from the web server to the client ensues according to the HTTP protocol, whereby the information transmitted to the client is read by the browser installed there and the individual tags are interpreted so that the information is displayed at the picture screen of the client in the predetermined fashion.

5 Since essentially only static images and texts that can be presented with the HTML format, the WWW service has been augmented by JAVA for incorporation of multimedia elements, animations or programs. JAVA is an object-oriented  
10 programming language for web applications developed by SUN Microsystems. JAVA supports text, hypertext, graphic, audio and animation functions.

15 The program packages programmed in JAVA can either be loaded onto the client from the web server as what are known as JAVA applets or can be pre-installed on the client as what are known as JAVA applications. A modification of JAVA that is referred to as JAVAscript [sic] allows the insertion of program packages into the HTML page.

20 JAVA applications with which both administration functions at the printers and printing systems as well as the sending of print jobs to the individual printers or, respectively, printing systems can be executed have been developed in order to control printers and printing systems via the Internet. These JAVA programs represent a significant advance relative to the previous situation since it enables the user of a client computer to control a plurality of printers and printing systems via  
25 the Internet, independent of platform. The type of the printing systems, i.e. the interface types, that can be controlled by a client are determined by the respective JAVA application, whereby a separate program element is provided for each interface type. This means that the more different interfaces that should be controlled with such a JAVA, the more extensive that this JAVA apparatus must  
30 be. This correspondingly applies for the functional range that should be controlled given such a JAVA application. This has the result that these JAVA applications

are extensive programs with, for example, a data measure of 9 MB. They thus demand considerable computing power and memory space at the client and are therefore expedient only for users who regularly wish to control different printers and printing systems.

5

It is also known to provide programs executable on the server that can be called by the client. These programs are either stored on the server as completely compiled programs or on the server as program code that can be interpreted by an interpreter. The programming language Perl is often employed for Internet applications for 10 such an interpretable program code. It is especially suited for databank applications to be executed at the server. However, pages that can be loaded from the server to the client can also be generated with Perl, whereby Perl provides commands with which the makeup of corresponding pages can be constructed. An HTML program code can also hereby be generated from the Perl program 15 code. The complete program code is interpreted and executed at the server given the call of such a program at the server. Programs based on Perl for controlling printers, printing systems and corresponding pre-processing and post-processing devices are not known to the inventor of the present invention.

20 A print server for professional, Internet-independent applications arises on pages 12-2 through 12-8 from "Das Druckerbuch" from Oc  Printing Systems GmbH, ISBN 3-00-001019-X, 3rd edition, October 1998. This print server is designated as PRISMApro®. Such a print server is composed of a high-performance personal computer with a corresponding software in order to be able to control one or more 25 printers, particularly a fast or high-performance printer. Such a print server is integrated into a network and converts the incoming data streams into corresponding print data. Dependent on the embodiment, the print server can process data in the formats AFDPS, Line/SF, PostScript, TIFF, PDF, LCDS, Line/O and PCL. Such print servers are in particular utilized in databank 30 applications, whereby a large data set with variable data is printed from a databank.

Furthermore, the print production control system Océ Domain® is described on pages 14-2 through 14-12 of “Das Druckerbuch” from Océ Printing Systems GmbH, ISBN 3-00-001019-X, 3rd Edition, October 1998. In this system one or more printers with pre-processing devices and post-processing devices are

5 connected by means of a network. The system controls and monitors the print production. It is a network-based client-server solution with a databank that is installed on a central databank server. All machine and operating data acquired in the production process are collected in this databank and provided to the various clients for later evaluations or can be exported into client-specific applications.

10 Interfaces to the high-performance printers connected on the network and to the various hardware and software components are realized with Océ Domain®. The industry standards DMI (**D**esktop **M**anagement **I**nterface), LMO (**L**arge **M**ailroom **O**perations) and ODBC (**O**pen **D**ata **B**ase **C**onnectivity) are supported. Océ Domain® is a high-performance control and monitoring system that is in particular

15 employed in printing centers in which high-performance printers are coupled with devices for pre-processing and for post-processing.

The invention is based on the object of finding a simple technical solution for the controlling of different printers and printing systems.

20 The object is achieved via a network with the features of the claim 1, an interpreter for such a network with the features of the claim 9 and via a method for operating a network with the features of the claim 15. Advantageous embodiments of the invention are specified in the sub-claims.

25 The network according to the invention is a client-server network that connects at least one client with at least one server, whereby files stored on the server can be retrieved by the client via communication of a file address, whereby a corresponding file is transferred to the server. These files that can be transferred

30 from the server to the client comprise language elements that are executed on the client.

The invention is characterized in that an interpreter is provided on the server, which interpreter can interpret and execute further language elements that can be executed on the server, which language elements are stored on the server and

5 contain files that can be retrieved by the client.

With the invention, files are thus for the first time provided on the server that comprise both language elements that can be executed on the server and language elements that can be executed on the client. Significant advantages are hereby

10 achieved since it is no longer necessary language elements executed on the client via language elements of another language (as it is, for example, known from Perl programs executable on the server), which is connected with significant programmer effort.

15 If the language elements executable on the client correspond to a markup language (such as, for example, SGML, HTML, XML), the files can be edited with the most common editors and text processing programs. Often no knowledge of the markup language is necessary to create simple markup files since the corresponding syntax is automatically inserted by the editor or, respectively, by the text processing

20 program. Complex applications can thus be produced with the invention using text processing programs that are simple to use. The language elements executable on the server must merely be added to the familiar files retrievable from the server. The combination of language elements executable on the client and language elements executable on the server in one file thus assures a significant facilitation

25 for the user who sets up the files at the server, since he can create these files with means with which he is familiar and need not encode the commands executable on the client in another computer language upon whose execution they are artificially generated.

30 Via the provision of an interpreter which examines the files called by the client for language elements contained therein and, if applicable, interprets and executes

these language elements on the server, the execution of an inherently arbitrary program on the server can be initiated by the transmission of only a single address from the client to the server. A possibility is thus achieved to be able to call arbitrary programs previously stored in corresponding files on the server with

5 minimal programming effort at the client and the most minimal data volume that must be transmitted between the server and the client.

These programs can be used by a plurality of clients, such that a corresponding program package need be stored at only a single point on the network (at the

10 server).

The advantages achieved with the invention become particularly clear when the data transmission in the network is based on a standardized data transmission such as, for example, that according to the HTTP protocol that is applied in the intranets and internets. Given employment of such a standard, only a traditional browser without further, additional programs is required at the client in order to control the corresponding functions. Previously, complicated programs (for example in JAVA) were loaded onto the clients (as was initially explained), whereby only the desired functions can be controlled at the clients on which the corresponding

15 programs have been installed or loaded. In contrast, given the inventive network the corresponding programs on the server can be called by every client on which a browser is present without additional programs having to be loaded onto the client. Only an access authorization to the respective server is required.

20 A significant functional advantage of the invention compared to the traditional system lies in that, according to the invention, it is possible that program elements that automatically connect the client to a further server are contained in the files stored on the server. This connection ensues, for example, simply by transmitting the address of the further server from the original server to the client, which

25 address is then executed [sic] on the client (i.e. is sent to further servers), whereby a connection is established between the further server and the client.

This connection function can also be designed as an automatic connection function, whereby a file is created at the server that contains the services of other servers, such that the client is automatically connected to a further server that can

5 attend to the service when a specific service is requested by a client at this one server and this server cannot fulfill the service. This function appears to the client as though his job were being negotiated between the servers in order to identify that server that can fulfill said job. This function of the automatic forwarding is therefore also designated as "trading".

10 The trading also comprises the possibility that the server functions as a type of transit or, respectively, relay station, whereby it switches a connection between the client and a further workstation such as, for example, a further server of a printing station or the like, given which the server is connected between the workstation

15 and the client and correspondingly forwards the data. If the server is provided with a gateway in such a case, the data transmission between the further workstation and the server and between the server and the client can be based on a different system or, respectively, different protocol, whereby the data are correspondingly translated by the server so that a frictionless data transmission is possible.

20 Given the storage of corresponding files on the server, a printing system with such a client-server network enables every client that can access this server to address the printer devices and pre-processing and post-processing devices that can be controlled by means of the server without a specific software for this having to be

25 installed on the client. The individual print jobs can also be forwarded from one server to another server in order, for example, to be printed out on a printer particularly suited for this specific print job. The server can be provided with gateways for the conversion of the data into specific print protocols or other types of transfer protocols so that printers connected to the network in the greatest

30 variety of ways can be controlled by means of a single server.

A printing station comprising at least one printer with, if applicable, a pre- and post-processing device can be connected to the network in a simple way in that the devices of the printing station (printer, pre-processing devices and post-processing devices) are connected with a server that is provided with the inventive interpreter

5 and on which corresponding programs for controlling the individual devices of the print station are stored in files that can be retrieved by the client. The devices of such a printing station can then be controlled by any arbitrary client over the network insofar as a corresponding access authorization to the server exists . The invention thus achieves a possibility with which a network access to the printing

10 station that can be addressed with extremely simple technical means is created merely by adding a server to an existing print station.

The inventive network is fashioned for controlling printers and corresponding pre-processing and post-processing devices. However, it can also be employed for

15 controlling any other devices since the basic principle of the present invention (the provision of an interpreter on a network server that interprets files called by the client) can be transferred to further, arbitrary network applications.

The invention is subsequently exemplarily explained in detail using the attached

20 drawings. Shown therein are:

Fig. 1 schematically , an inventive network in a block diagram,

Fig. 2 an interface query in a flow chart that shows the program execution

25 of the computer program attached as an appendix and

Fig. 3 a trading method in the flow chart.

Figure 1 shows an exemplary embodiment of an inventive network. The network

30 comprises a first web server 1, a second web server 2 and a first client and second client 3, 4. The first and second web server and the first client are connected with

one another via data lines 5, 6 by means of a LAN, whereby the data line 5 connects the first web server with the first client and the data line 6 connects the first web server 1 with the second web server 2. An intranet is operated on the LAN.

5

No fixed, physical data connection exists between the first web server 1 and the second client 4. A corresponding data connection 7, for example via the telephone network, is established as needed, whereby the Internet is employed as transmission medium.

10

The first client 3 is, for example, an office computer of a user of the inventive network (which office computer is connected with the first web server 1 via the intranet), in contrast to which the client 2 is a personal computer situated at the user's home with which the user can dial into the Internet by means of a modem and establish a connection to the first web server 1. Since the data connection 7 is not permanently present, it is drawn dashed in Figure 1.

A browser is respectively installed on both clients 3, 4 so that both clients 3, 4 can communicate with the web server 1 with the services known from the Internet.

20

These services are, for example, Telnet (which) enables a terminal simulation or FTP (with which files can be transmitted). The service of the World Wide Web (WWW), which automatically resorts to the further services (such as, for example, FTP, news, Telnet, Gopher, e-mail, etc.) of the Internet as needed, is used in the present exemplary embodiment. Two devices 8, 9 are connected to the first web server 1, whereby the device 8 is connected with the web server 1 by means of a serial line (RS 232/V24) and a connection 11 according to the SNMP protocol (Simple Network Management Protocol) is installed between the device 9 and the server 1. The SNMP protocol is in particular employed to control devices connected to a network. At the web server 1 a gateway 12 is provided that converts the data arriving via the SNMP connection 11 from the SNMP protocol into the Internet protocol applied by the web server 1 on the data connections 5

through 7 or, respectively, conversely converts the data according to the Internet protocol into the SNMP protocol when they are transmitted from the web server 1 to the control device 9. All hardware and software components at the web server that establish a connection to a communication system or, respectively, network  
5 different from the Internet are a gateway in the sense of the invention. A plurality of further gateways to other communication systems that are, for example, based on the DMI standard (**Desktop Management Interface**), the LP standard (**Line Printer**), SLP (**Service Location Protocol**), the IPP standard (**Internet Printing Protocol**) can be provided on the web server 1. Gateways to the PJMweb are  
10 possible. PJMweb is a web-based print client that has been developed by Oc . A gateway can also generate a terminal emulation. In particular, gateways can be provided that are fashioned to control specific printer communication systems.

Since a connection to different networks can be established via the gateways,  
15 arbitrarily many devices can in principle be connected to a server provided with a gateway or, respectively, can be controlled by the server.

The second web server 2 is connected with a further device 13 via a line 14. For this a communication program 15 specifically fashioned for the device 13 is  
20 installed on the second server 2. This communication program 15 is a compiled program for what is known as web-based management. Such programs are respectively configured for a specific application such as, for example, controlling the device 13, whereby only a single connection to the device 13 can be established and only a specific device type can be controlled with this program.

25 In addition to other service programs, the first and second web server 1, 2 comprise a WWW service program 16 that exhibits the functions known from the prior art, meaning that it reads the corresponding pages out from a data store 17 of the web server 1, 2 upon receipt of a URL and sends them to the client 3, 4 that sent the  
30 URL. According to the invention, an interpreter 18 is provided on the server 1, 2 with which the pages deposited in the store 17 are interpreted before being sent to a

client 3, 4, meaning that language elements contained in the pages are executed by the interpreter 18. The individual pages are stored in the HTML format, meaning that they comprise standardized language elements that can be executed by the browser of the client 3, 4. The language elements executable by the interpreter 18  
5 are independent of the language elements executable by the browser. They can also be viewed as a supplement to the HTML format, for which reason the format of the stored pages can also be referred to as an extended HTML format.

A program code of a program example that is explained in detail below is specified  
10 in the appendix. The scope of the commands of the interpreter is initially explained in brief. Like most computer languages, the interpreter comprises commands for string processing, system commands and structure commands. As needed, the interpreter can be provided with further groups of commands. In addition to the aforementioned groups of commands, the interpreter comprises a  
15 command with which new commands of the interpreter can be generated. This command is "addfunction", with which an arbitrary command (function) can be constructed from arbitrary commands (machine language, operating systems or other high level languages) available on the server. "addfunction" is not used in the HTML pages but can be called by the user in a program development  
20 environment.

Important string processing commands are "userDefineString",  
"userGetCGIString", "userPreReplaceString", "userPostReplaceString" and  
"userCompose". String variables can be defined with "userDefineString". A string  
25 input at the client can be imported with "userGetCGIString". The commands  
"userPreReplaceString" and "userPostReplaceString" serve to arrange a string at a predetermined location of the page, whereby with the command  
"userPreReplaceString" the string is arranged at the corresponding location immediately after the execution of this command, in contrast to which the string is  
30 only arranged at the predetermined location after the processing of all language elements given "userPostReplaceString". The string represented by the command

"userPreReplaceString" can thus still be modified by the further language elements to be executed by the interpreter, in contrast to which the string arranged with the command "userPostReplaceString" is placed at the end, whereby a modification of the string is no longer possible.

5

One of the most powerful system commands is "userSystem (...)", whereby a command of the operating system of the server or of a further program installed on the server can be input into the parentheses. Commands and programs installed on the server can hereby be called with the interpreter. This command is called from 10 an HTML page.

A sub-group of system commands are control commands that, for example, serve to control a specific printer. These commands usually correspond to the respective printer system commands that are respectively converted as a command of the 15 interpreter and can be supplemented by further, higher control commands.

The structure commands serve to create a program structure with branchings, loops and the like. Corresponding structure commands are, for example, "userFor", "userGoSub" or "userIf".

20

Corresponding to the syntax of HTML, the commands executable by the interpreter 18 are also placed in "<...>".

Given the commands of the interpreter 18 a distinction is made between two 25 classes, namely one with commands that cannot be directly called by the client and a further class whose commands can be directly called by the client. Of the commands specified above by way of example, those that begin with "user" cannot be called by the client. This division into two classes of commands serves for security because, if a user of an arbitrary client could generate a "userSystem" 30 command on the server, he could manipulate the server at will. To avoid such a manipulation, the store 17 is subdivided into a free memory area 17a and a locked

memory area 17b. In the free memory area pages can be called by the client and these are, if applicable, supplied with parameters, in contrast to which the locked memory area 17b contains pages that do not receive parameters directly from the client or that cannot be called by said client. These pages are only called by the  
5 interpreter. The interpreter thereby monitors the parameter handover and the program execution. The interpreter 18 is fashioned such that security-relevant commands are only executed when they are stored in the locked memory area 17b. A user at one of the clients can only read and directly address the content of the free memory area 17a. The pages deposited in the locked memory area 17b can  
10 only be addressed indirectly via the interpreter.

The administrator of the server can deposit control programs needed to control the devices 8, 9 and 13 in the locked memory area 17b of the server, which control programs are usually security-relevant.

15 The program code specified in the appendix is subsequently explained in detail, the program execution of which is shown in Figure 2 in a flow chart. An initialization is implemented in the step S1 "ini", whereby specific values and strings are set or, respectively, defined for respective servers to be queried.

20 The SNMP gateway 12 is called with the step S2, whereby via the value "1" it is specified that this something should be imported by this gateway, whereby the result is stored in "SNMPVALUE". Dependent on a user input, an address of a databank entry (managed object of an MIB) is hereby queried. The result of this  
25 query represents the computer type that is addressed via the SNMP interface.

Using the value stored in the variable "SNMPVALUE", in the next step S3 it is checked whether the operating system of the computer addressed via the SNMP gateway is Windows. In the event that the operating system is Windows, the  
30 program execution branches to the query S4 in which an SNMP value is imported again and, using this value, it is checked whether a predetermined printing system

("Imagestream") that can be run under Windows is set up on the computer. When the query S4 yields the presence of such a printing system, this is stored in the step S5.

- 5 After the storing event in the step S5 or, respectively, in the event that one of the queries S3 or S4 was answered in the negative, the program workflow transitions to the step S6 with which it is checked whether the operating system of the computer addressed via the SNMP gateway is a Unix operating system. If the query yields that the operating system is a Unix operating system, the program execution branches to the step S7 with which an SNMP value is imported again using which it is checked whether a Unix operating system (PJM or, respectively, Prisma) is set up on the computer. When the query yields such a printing system, then this is stored in the step S8.
- 10
- 15 After the storing event of step S8 or when one of the two queries of the steps S6 and S7 had a negative outcome, the program execution passes to a step S9 with which it is checked whether the computer addressed via the SNMP gateway is a computer that can be controlled at all by means of the SNMP protocol. If the query yields that the computer can in principle be controlled by means of the
- 20 SNMP protocol, a PING query is called in step S10 and the result of the PING query is checked in step S11 as to whether the computer is operating (online). If this query S11 yields that the computer is operating (online), this is stored in step S12; contrarily, when the computer is not operating (offline), this then corresponds to the default setting such that no additional storing is necessary.
- 25
- 30 The PING query is a program module set up on many servers and present not only on the Internet. It is therefore called with the "userSystem" function already described above or is integrated into the interpreter as an independent command. A reprogramming of this program module is thus superfluous.

After the storing event of the step S12, or in the event that the query in step S9 yielded that the computer is an SNMP computer, or in the event that the query in step S11 yielded that the computer is not operating, the program execution passes to the step S13 with which an image representing the printing system is registered.

- 5 This ensues with the command "userCompose" with which the variable "SNMPVALUE" documented by the above queries and storing events is evaluated (see the line under the command "userCompose"), whereby the result is "exclusive". This is prescribed in the following line by the value "false". This means that only a single image can be read into the variable "REPLACEMENT".
- 10 Depending on the above result, an image for a [sic], an Imagestream printing system (imagestream.gif); a Prisma printing system (Prisma.gif); or an image for a successful PING query (ping.gif); or an image for the non-operational status of the computer (offline.gif); or an image for a computer addressable by means of the SNMP protocol (SNMP.gif) is imported.
- 15 In the following steps S14 and S15 a hyperlink start variable ("SnmpHLStart") or, respectively, a hyperlink end variable ("SnmpHLEnd") with the strings needed for generation of a hyperlink is again documented by means of the command "userCompose". A hyperlink is an automated call of a file of the server, whereby a client sends the corresponding URL to the server comprising the file.

- 20 A program arrow from step S15 to step S2 that closes the program section between the steps S2 and S15 into a loop is drawn dashed in Figure 2. With this program arrow it is expressed that this program section is executed multiple times to query a plurality of computers reachable via the SNMP gateway. The queries of the individual computers are executed in parallel. This is possible because the SNMP gateway is capable of multitasking.

- 25 In the following step S16 the strings representing the hyperlink are inserted into the file or, respectively, HTML page at a predetermined location by means of the command "userPostReplaceString".

With the above-described program it is thus queried whether one or more specific computers can be addressed with the SNMP gateway and whether a specific printing system is installed on one of the addressable computers. A corresponding  
5 image is then imported and a link pointing to the computer is generated and deposited in the HTML page. If the HTML page is now transmitted to the client, the link can be either manually or automatically executed, whereupon a connection is established from the client that has started the query directly to the computer with the queried printing system.

10

The client was thus switched from one server to a further server, whereby the first server sought for the client a printing system with which corresponding print jobs can then be executed directly by the client. The search and switching procedure has been completely executed at the server; this shows that the necessary  
15 "intelligence" must be provided only at the server and can be queried and operated by the client with a traditional browser.

20 The above program is merely a severely simplified and abbreviated example for a query of reachable printing systems and automatic or, respectively, semi-automatic switching to a desired printing system. This example should merely indicate which possibilities are created via the inventive provision of an interpreter on the server.

In Figure 3 a flowchart is presented with which it is shown how a print job can be executed with the inventive network. In a step S18 the user inputs the print job and  
25 augments this with specific requests for the quality of the printed product (color, paper type etc.) and necessary print features (number of copies, format etc.).

In step S19 this print job is transferred from the client 3, 4 to the server 1, 2. At the server a database that contains the relevant data for connections to printers or,  
30 respectively, to further servers with connected printers or, respectively, corresponding pre-processing and post-processing devices is updated (S20). The

updating S20 ensues in that one or more gateways are queried for corresponding printing systems. The data hereby determined are entered in the local database stored on the server.

5 This updated database is evaluated in step S21 according to the parameters input by the user (print job, requests and requirements).

With a query S22 it is subsequently checked whether a printer device suitable for the print job has been found in the evaluation. In the event that no suitable printer device has been found, the program execution branches to the step S23 with which a message that the print job cannot be executed is sent to the client.

Contrarily, if the query in step S22 yields that a suitable printer device is present, with a further query S24 it is checked whether the print job can be executed by the server. If the result of this query is no, the client is switched S25 to a further server that can execute the print job, as was shown using the program example from Fig. 2. The print job together with the parameters (requests and requirements) is hereby advantageously directly transmitted to the further server, whereby the parameters can be correspondingly modified as needed.

20

If, in contrast, the query S24 yields that the print job can be executed by the present server, in a further query S26 it is checked whether the print job can be directly communicated to the printer device. If the printer device possesses a corresponding network coupling, a direct link to the printer device can be generated, whereby the server is no longer burdened by the print job.

If the query from step S26 yields that a direct communication of the print job to the printer device is possible, this is executed in step S27.

30 If such a communication is not possible, a link to the server or, respectively, its gateway is set with the step S28 in order to transmit the print data to the printer

device via the gateway in step S29. After the end of the printing event, a print confirmation S30 ensues from the server to the client, whereupon the method is ended.

- 5 In this method the print job is automatically forwarded to a suitable printer device. Who is available and suited for execution of the print job is, so to speak, negotiated between individual servers and printer devices. Such a method is therefore also designated as "trading".
- 10 As is presented above, the inventive network is particularly suited for the formation of a decentralized printing system that can comprise one or more print servers. Given an Internet application of the print server, each Internet client that possesses a corresponding access authorization to the server can use this for its print jobs. The technical realization of such a decentralized printing system is
- 15 extremely simple and merely requires the installation of an inventive interpreter in which the corresponding commands for controlling the printer devices or, respectively, the pre-processing and post-processing devices are set up. A plurality of servers can be provided with an inventive interpreter in a single network. It is hereby also possible that a print job is handed off between a plurality of servers.
- 20 However, the invention is not limited to a printing system but rather can be employed for controlling, monitoring, maintenance etc. of arbitrary devices. Currently there thus exist considerable efforts to make household appliances network-capable. With an inventive server they can be checked, queried and
- 25 potentially placed into operation by a user via the Internet proceeding from an arbitrary location. In principle the administration and control of all technical devices is possible with the inventive network. In particular it is suitable for the administration of data processing systems, telecommunication systems and switching systems, whereby it is especially advantageous for super-regional
- 30 systems since the inventive server can be controlled from an arbitrary location of the network.

Another significant advantage of the inventive network is that , due to the provision of an interpreter, it is not limited to a specific application; rather, due to the design possibility of a computer language it is possible to generate the

5       respective application with language elements similar to a high-level language. The inventive system hereby possesses a maximum flexibility. For most applications there are already specialized program parts that merely have to be integrated into the interpreter. Such program parts are, for example, gateways, printer drivers, spoolers or other unique control programs. In particular,  
10      specialized communication protocols such as SNMP or DMI can be utilized for the local data transfer to the printer devices since these protocols – in contrast to the HTTP protocol – possess a significantly lower protocol overhead, shorter response times and a high performance and allow a simple application. These specialized protocols or, respectively, specific interfaces can be made accessible to a user who  
15      needs neither deal with this technology nor install a corresponding, usually very complicated software on his client in order to be able to establish a communication with such specialized systems.

20      The inventive interpreter can be stored on a data medium and be loaded into a server from this or via a network.

The invention can be briefly summarized according to the following:

25      The present invention concerns a client-server network, an interpreter installable on the server of the network and a method for operation of such a client-server network.

30      The invention is characterized in that files are deposited at the server, which files can be retrieved by the client and inventively comprise both language elements executable at the client and language elements executable at the server. Provided

on the server is an interpreter that interprets and executes the language elements executable at the server.

According to a preferred embodiment of the invention, the language elements  
5 executable at the client correspond to a markup language (such as, for example, SGML, XML, HTML) since the user, upon establishing these files, can then use his known aids to create the files (which aids are usually familiar text processing programs) in order to provide individual applications on the server of the network that can be called by an arbitrary client with a traditional browser.

10

The invention is particularly suited for the control of devices, particularly of printers and printing systems and the corresponding pre-processing and post-processing devices, since the control intelligence is centrally stored at the server and can thus be used by many clients and the data transfer between the  
15 clients and the server is kept low.

Another aspect of the invention is that a trading of, for example, print jobs between a plurality of servers can be realized with simple means.

## **Reference list**

1	first web server
2	second web server
5	3 first client
	4 second client
	5 data line (intranet)
	6 data line (intranet)
	7 data connection (Internet)
10	8 device
	9 device
	10 serial line
	11 SNMP connection
	12 gateway
15	13 device
	14 line
	15 communication program
	16 WWW service program
	17 storage
20	17a free memory area
	17b locked memory area
	18 interpreter

### **Method steps**

- S1 initialization
- 5 S2 call and query of the SNMP gateway
- S3 Windows?
- S4 Windows printing system?
- S5 storing
- S6 UNIX?
- 10 S7 UNIX printing system?
- S8 storing
- S9 no SNMP?
- S10 PING
- S11 PING?
- 15 S12 storing
- S13 UserCompose: image
- S14 UserCompose: hyperlink start
- S15 UserCompose: hyperlink end
- S16 insertion of the hyperlink into HTML page
- 20 S17 end
- S18 input of the print job
- S19 transfer from client to server
- S20 updating of a database
- S21 evaluation of the database
- 25 S22 suitable printer device?
- S23 message to client
- S24 can server execute print job?
- S25 switching to further server
- S26 can communication be carried out to printer device?
- 30 S27 communication to printer device with direct link
- S28 link to server

S29            transmission via gateway  
S30            print confirmation

## **Claims**

1. Network of the client-server type that connects at least one client (3, 4) with at least one server (1, 2), whereby files stored on the server (1, 2) are transferred from the server (1, 2) to the client (3, 4) when the client (3, 4) retrieves them by sending a corresponding file address to the server (1, 2) and  
5 the files contain both language elements executable at the client (3, 4) and language elements executable at the server (1, 2) and  
10 an interpreter for interpretation and execution of language elements executable at the server is present on the server (1, 2).
2. Network according to claim 1,  
characterized in that  
15 the interpreter (18) on the server (1, 2) is fashioned such that the language elements executable at the server are executed at the server (1, 2) after the calling of the files by a client and before the transmission of the files to the client (3, 4).
- 20 3. Network according to claim 1 or claim 2,  
characterized in that  
the file address corresponds to the URL format and the server (1, 2) is a web server, such that the files can be called with an Internet browser installed on the client (3, 4).  
25
4. Network according to any of the claims 1 through 3,  
characterized in that  
the files stored on the server (1, 2) and retrievable by the client (3, 4)  
correspond to the format of a markup language (SGML, XML, HTML) that  
30 is extended by the language elements executable at the server.

5. Network according to any of the claims 1 through 4,  
characterized in that  
at least one gateway (12) is installed on the server (1, 2), which gateway  
(12) can establish a data connection to a further logical and/or physical  
5 system (device, network, queue, computer), whereby the data of the further  
system exhibit a different format than the data exchanged between the  
server (1, 2) and the client (3, 4), and the gateway (12) automatically  
converts both the incoming and the outgoing data into the corresponding  
data formats.

10

6. Network according to claim 5,  
characterized in that  
a gateway (12) is respectively provided for conversion of the data into one  
or more of the following formats: SNMP, LP, PJMweb, ftp.

15

7. Network according to claim 5 or 6,  
characterized in that  
the gateway or, respectively, gateways (12) are integrated into the  
interpreter and can be called by language elements of the interpreter (18).

20

8. Network according to any of the claims 1 through 7,  
characterized in that  
programs for controlling at least one printer and/or pre-processing or  
post-processing devices (such as, for example, of a printer driver or of a  
25 spooler) are installed on the server (1, 2) and these programs can be called  
by the interpreter (18).

9. Interpreter for a network, particularly according to any of the preceding  
claims, that can be installed on a server (1, 2) of a client-server network and  
is fashioned for interpretation and execution of language elements  
30 executable at the server (1, 2) that are contained in files stored on the server

(1, 2), whereby these files can be called by a client (3, 4) by means of the transmission of an address and contain additional language elements executable at the client (3, 4).

5      10. Interpreter according to claim 9,  
          characterized in that  
          the interpreter comprises a command (userCompose) for generating string  
          entries in the file.

10     11. Interpreter according to claim 9 or 10,  
          characterized by  
          a command (userPreReplaceString, userPostReplaceString) for setting  
          string entries at a predetermined location of the file.

15     12. Interpreter according to any of the claims 9 through 11,  
          characterized by  
          a command for importing a string transmitted from the client (3, 4) to the  
          server (1, 2) and for storer [sic] the string in a predetermined variable  
          (userGetCGIString).

20

13. Interpreter according to any of the claims 9 through 12,  
          characterized by  
          a command for calling a gateway and querying a system connected with the  
          gateway.

25

14. Interpreter according to any of the claims 9 through 13,  
          characterized in that  
          the interpreter comprises a group of client commands that can be called  
          both from the client as well as from the server and comprises a group of  
          sever commands that can only be called from the server.

30

15. Interpreter according to any of the claims 9 through 14,  
characterized in that  
the interpreter is stored on a data medium.
  
- 5      16. Method for operation of a network according to any of the claims 1 through  
8, whereby  
the network is a client-server network and files stored on the server (1, 2)  
are transferred from the server (1, 2) to the client (3, 4) when the client (3,  
4) retrieves them by sending a corresponding file address to the server (1,  
10     2), and the server respective offers one or more services to the clients,  
whereby,  
given a client request for a specific service with specific parameters  
forming the basis of the service, the queried server determines whether it  
can fulfill the service, and when the server establishes that it cannot fulfill  
15     the service, it connects a further server or a device connected to a network  
to the client, which further server or device can execute the service.
  
17. Method for operation of a network according to claim 16,  
characterized in that
  
- 20     one of the services offered by the servers is the execution of a print job and  
the server forwards the print job to a other server or directly to a printer  
device when the server itself cannot execute the print job.
  
- 25     18. Method for operation of a network according to claim 16 or 17,  
characterized in that  
the server comprises a databank in which information regarding the  
services offered on the network are stored so that, given a client request,  
whether the desired service is present on the network can be determined  
using this databank.

19. Method for operation of a network according to any of the claims 16 through 18,  
characterized in that  
the switching to a further server or to a device connected to the network is  
5           executed by generating the address of the further server or of the device and  
by communicating the address to the querying client.
  
20. Method for operation of a network according to any of the claims 16 through 19,  
10           characterized by  
an interpreter according to any of the claims 9 through 15.

## Abstract

The present invention concerns a client-server network, an interpreter installable on the server of the network and a method for operation of such a client-server  
5 network.

The invention is characterized in that files are deposited at the server, which files can be retrieved by the client and inventively comprise both language elements executable at the client and language elements executable at the server. Provided  
10 on the server is an interpreter that interprets and executes the language elements executable at the server.

According to a preferred embodiment of the invention, the language elements executable at the client correspond to a markup language (such as, for example,  
15 SGML, XML, HTML) since the user, upon establishing these files, can then use his known aids to create the files (which aids are usually familiar text processing programs) in order to provide individual applications on the server of the network that can be called by an arbitrary client with a traditional browser.

20 The invention is particularly suited for the control of devices, particularly of printers and printing systems and the corresponding pre-processing and post-processing devices, since the control intelligence is centrally stored at the server and can thus be used by many clients and the data transfer between the clients and the server is kept low.

25

Another aspect of the invention is that a trading of, for example, print jobs between a plurality of servers can be realized with simple means.

(Figure 2)

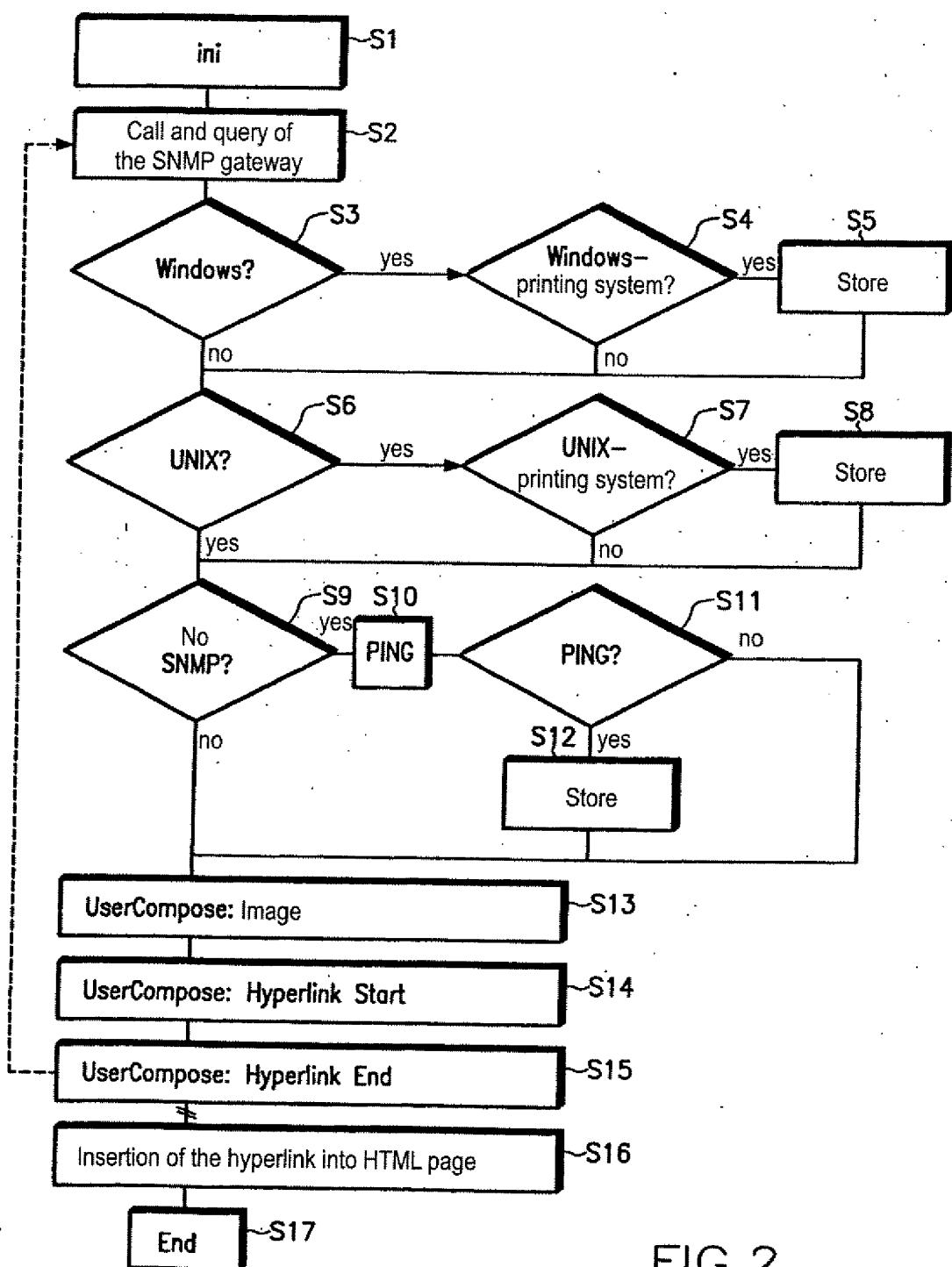


FIG.2

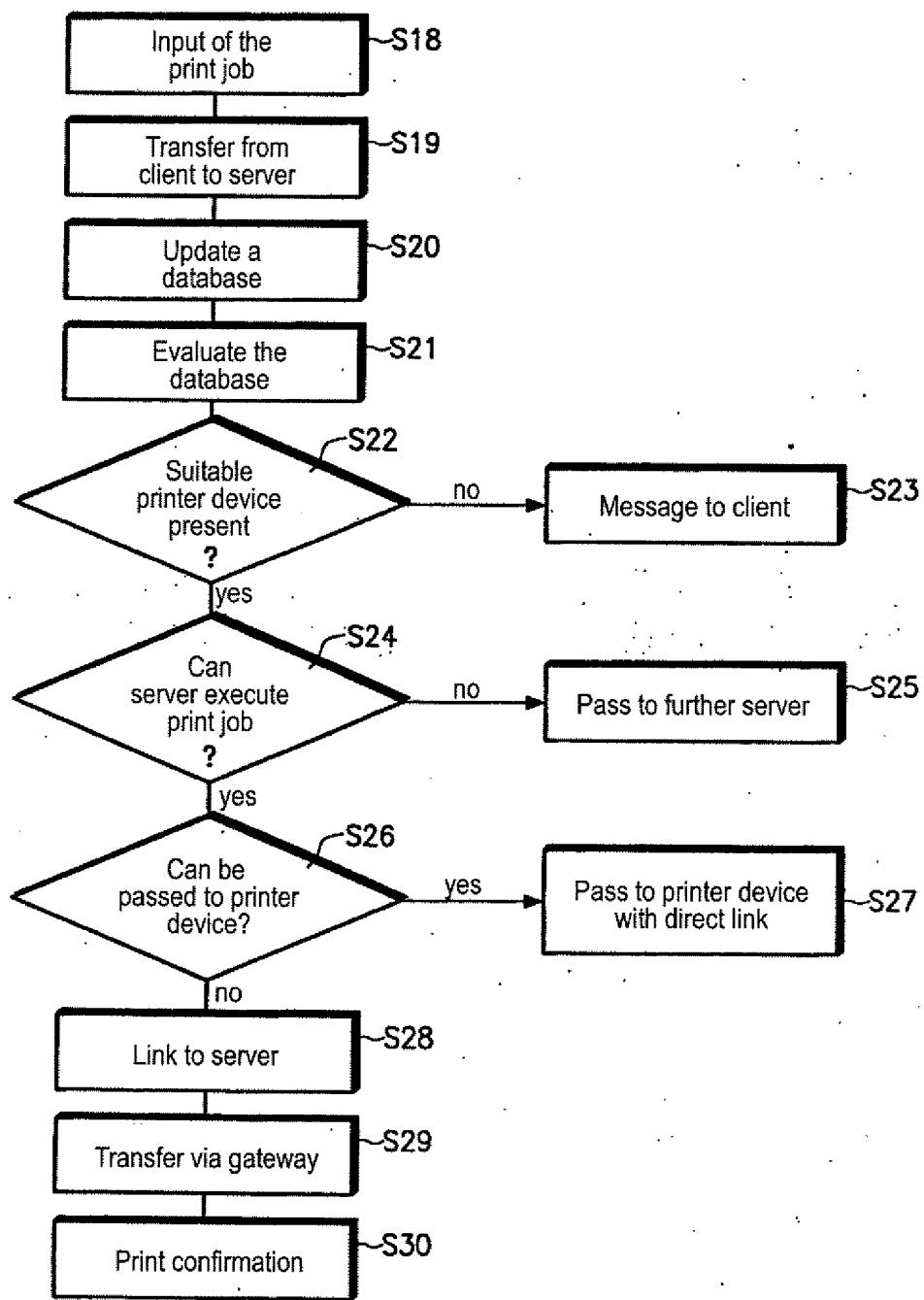


FIG.3

## Appendix

```

<head>
<TITLE>SNMP Response</TITLE>
</HEAD>

5 <BODY>

10 <!--
----- Commentary: program start
-----//-->

15 <!--
----- Commentary: function "STDefineString": is called by init
-----//-->

20 <PSUB FUNC="STDefineString">
<PTAG FUNC="userDefineString" VALUE="SNMPVALUE$i%" VALUE="Offline"></PTAG>
<PTAG FUNC="userDefineString" VALUE="REPLACEMENT$i%"></PTAG>
<PTAG FUNC="userDefineString" VALUE="ShmpHLStart$i%"></PTAG>
25 <PTAG FUNC="userDefineString" VALUE="ShmpHLEnd$i%"></PTAG>
<PTAG FUNC="userDefineString" VALUE="Stdout$i%"></PTAG>
<PTAG FUNC="userDefineString" VALUE="Stdin$i%"></PTAG>
<PTAG FUNC="userDefineString" VALUE="Stderr$i%"></PTAG>
30 </PSUB>

<!--
----- Commentary: function "MTScanSystem"
-----//-->

35 <PSUB FUNC="MTScanSystem">
<PTAG FUNC="userSNMPGateway" // Commentary: Section s2
40   VALUE="SNMPVALUE$i%" VALUE="1"
     VALUE="160.120.17.$i%"
     VALUE="public"
     VALUE=".1.3.6.1.2.1.1.1.0">
45 </PTAG>

<PBRANCH FUNC="userIf" // Commentary: Section s3
50   VALUE="SNMPVALUE$i%" VALUE="FNC>WIN" // Commentary: Section s4
     VALUE=<PTAG FUNC="userSNMPGateway" // Commentary: Section s5
       VALUE="SNMPVALUE$i%" VALUE="1"
       VALUE="160.120.17.$i%"
       VALUE="public"
55     VALUE=".1.3.6.1.4.1.1552.92.2.0">
     </PTAG>
</PBRANCH>

60 <PBRANCH FUNC="userIf" // Commentary: Section s6
    VALUE="SNMPVALUE$i%" VALUE="FNC>SCO" // Commentary: section s7
    VALUE=<PTAG FUNC="userSNMPGateway" // Commentary: section s8
      VALUE="SNMPVALUE$i%" VALUE="1"
65           VALUE="1"

```

```

                                VALUE="160.120.17.%i"
                                VALUE="public"
                                VALUE=".1.3.6.1.3.1.1.1.1.0">
                </PTAG>
5     </PBRANCH>

                <PBRANCH FUNC="userIf"                                // Commentary: Section s9
                                VALUE="SNMPVALUE%i"
                                VALUE="FNC>Offline"
10      VALUE=<PTAG FUNC="userSystem"                      // Commentary: Section s10
                                VALUE="Stdout%i"
                                VALUE="Stdin%i"
                                VALUE="Stderr%i"
                                VALUE="0"
15      VALUE="5000"
                                VALUE="ping -n 1"
                                VALUE="160.120.17.%i">
                </PTAG>
20      </PBRANCH>

                <PBRANCH FUNC="userIf"                                // Commentary: Section s11
                                VALUE="Stdout%i"
                                VALUE="FNC>Antwort"
                                VALUE=<PTAG FUNC="userDefineString" // Commentary: Section s12
                                VALUE="SNMPVALUE%i"
                                VALUE="PING">
                </PTAG>
25      </PBRANCH>

30      <PTAG FUNC="userCompose"                            // Commentary: Section s13
                                VALUE="SNMPVALUE%i"
                                VALUE="false"

                                VALUE="REPLACEMENT%i"
35      VALUE="FNC>ISTREAM"
                                VALUE="../images/imagesstream.gif"

                                VALUE="REPLACEMENT%i"
                                VALUE="FNC>PRISMA"
40      VALUE="../images/prisma.gif"

                                VALUE="REPLACEMENT%i"
                                VALUE="FCS>PING"
                                VALUE="../images/ping.gif"
45      VALUE="REPLACEMENT%i"
                                VALUE="FNC>Offline"
                                VALUE="../images/offline.gif">
                </PTAG>
50      <PTAG FUNC="userCompose"                            // Commentary: Section s14
                                VALUE="SNMPVALUE%i"
                                VALUE="false"

55      VALUE="SnmpHLStart%i"
                                VALUE="FNC>ISTREAM"
                                VALUE=<a HREF="\http://www.ops.de\">">
                                VALUE="SnmpHLStart%i"
60      VALUE="FNC>PRISMA"
                                VALUE=<a HREF="\http://160.120.17.%i/pjm.html\" tar-
get="\_blank\">>
                                VALUE="SnmpHLStart%i"
65      VALUE="FNC>Offline"
                                VALUE="">
                </PTAG>

```

```

        <PTAG FUNC="userCompose"                                // Commentary: Section s15
          VALUE="SNMPVALUE%1%" 
          VALUE="false"
5           VALUE="SnmpHLEnd%1%" 
          VALUE="FNC>ISTRAM" 
          VALUE="</a>" 

10          VALUE="SnmpHLEnd%1%" 
          VALUE="FNC>PRISMA" 
          VALUE="</a>" 

          VALUE="SnmpHLEnd%1%" 
15          VALUE="FNC>Offline" 
          VALUE="">
        </PTAG>
</PSUB>

20 <!--
----- Commentary: Function "STPrintSystem": is called by s16
-----//-->
25 <PSUB FUNC="STPrintSystem">
  <PTAG FUNC="userPreReplaceString" VALUE="SnmpHLStart%1%"></PTAG>
  <img src=<PTAG FUNC="userPreReplaceString" VALUE="REPLACEMENT%1%"></PTAG>
width="80" height="80"
30   alt="160.120.17.%1"
  <PTAG FUNC="userPreReplaceString" VALUE="SNMPVALUE%1%"></PTAG>">
  <PTAG FUNC="userPreReplaceString" VALUE="SnmpHLEnd%1%"></PTAG>
</PSUB>

35 <!--
      Kommentar: ini1 - s1
-----//-->
40 <PTAG FUNC="userSetSNMPRetries" VALUE="2"></PTAG>
<PTAG FUNC="userSetSNMPTimeout" VALUE="2000"></PTAG>

        <PTAG FUNC="userFor"
          VALUE="i"
          VALUE="200"
45          VALUE="255"
          VALUE="1"
          VALUE="0"
          VALUE="STDdefineString">
        </PTAG>
50

<!--
----- Commentary: Steps s2-s15 → call of the corresponding
55     sub-programs of sections s2-s15
-----//-->

        <PTAG FUNC="userFor"
          VALUE="i"
          VALUE="200"
          VALUE="255"
          VALUE="1"
          VALUE="-1"
65          VALUE="MTScanSystem">
        </PTAG>

```

```
<!--
-----
Commentary: Step s16
-----
5 //-->

<p>
<PTAG FUNC="userFor"
      VALUE="i"
      VALUE="200"
      VALUE="255"
      VALUE="1"
      VALUE="0"
      VALUE="STPrintSystem">
10   </PTAG>
</p>

<!--
-----
20   Commentary: Step s17 / end
-----
//-->

25   </BODY>
      </HTML>
```

PCT 00/04312

BUNDESREPUBLIK DEUTSCHLAND

EP00/4312

10/009397



REC'D 07 JUL 2000

WIPO

PC

EJU

Prioritätsbescheinigung über die Einreichung  
einer Patentanmeldung

Aktenzeichen: 199 22 118.9  
Anmeldetag: 12. Mai 1999  
Anmelder/Inhaber: Océ Printing Systems GmbH,  
Poing/DE  
Bezeichnung: Netzwerk, Interpreter für ein derartiges Netzwerk und  
Verfahren zum Betreiben eines Netzwerkes  
IPC: G 06 F 15/173

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Anmeldung.

München, den 02. Juni 2000  
Deutsches Patent- und Markenamt  
Der Präsident  
Im Auftrag

A handwritten signature in black ink, appearing to read "Dzierzon".

Dzierzon

**PRIORITY  
DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

Océ Printing Systems GmbH

990503 DE

10.05.99

**Netzwerk, Interpreter für ein derartiges Netzwerk**

**5 und Verfahren zum Betreiben eines Netzwerkes**

Die Erfindung betrifft ein Netzwerk, einen Interpreter für ein  
derartiges Netzwerk und ein Verfahren zum Betreiben eines Netz-  
10 werkes. Insbesondere betrifft die Erfindung ein Netzwerk für  
ein Drucksystem.

An den bestehenden Datennetzen, wie z.B. LANs (Local Area Net-  
works) und WANs (Wide Area Networks) sind in der Regel mehrere  
15 Drucker und Drucksysteme und eventuell entsprechende Vor- und  
Nachbearbeitungsgeräte, die z.B. das bedruckte Papier auf ein  
spezielles Format zurechtschneiden oder binden, angeschlossen.  
Die einzelnen Drucker und Drucksysteme unterscheiden sich in  
ihrer Leistungsfähigkeit stark, so können z.B. kleine Tinten-  
20 strahldrucker mit einer Druckleistung von 4 Seiten pro Minute  
oder Hochleistungsdrucker mit einer Druckleistung von 40 Seiten  
oder mehr pro Minute an das Datennetz angeschlossen sein. Die  
Drucker und Drucksysteme unterscheiden sich nicht nur in ihrer  
Druckleistung, sondern auch in der Druckqualität. So gibt es  
z.B. Drucker die nur in einer einzigen Farbe (monochrom) druk-  
ken, wohingegen auch zunehmend Farbdrucker zum Einsatz kommen.  
Elektrophotographische Hochleistungsdruckgeräte sind heutzutage  
ohne weiteres in der Lage, in zwei Farben zu drucken, das  
heißt, in einem sogenannten Spot-Color-Betrieb oder Highlight-  
30 Color-Betrieb. Ein derartiger Drucker ist beispielsweise von  
der Océ Printing Systems GmbH unter der Bezeichnung PAGESTREAM®  
200DSC bekannt.

Die unterschiedlichen Drucker und Drucksysteme sind in der Re-  
35 gel auch mit unterschiedlichen Systemschnittstellen an das je-  
weilige Netzwerk gekoppelt, wie z.B. dem SNMP (Simple Network  
Management Protokoll) oder dem DMI (Desktop Management Inter-

face). Obwohl oftmals über das Netzwerk zu mehreren Druckern bzw. Drucksystemen eine physikalische Verbindung besteht, kann nur ein Teil der Drucker angesprochen werden.

5 Mit der rasanten Entwicklung der Intra- und Internets, mit welchen mehrere LANs und WANs zu einem einzigen Netz verbunden sind, insbesondere durch die Einführung des auf dem HTTP-Protokoll (**Hypertext Transport Protokoll**) basierenden WWW-Dienstes, steigt die Anzahl der Drucker und Drucksysteme, die  
10 von einem einzigen Benutzer grundsätzlich angesprochen werden können explosionsartig an, wobei auch die Vielfalt der Protokolle zum Ansprechen der Drucker und Drucksysteme entsprechend zunimmt.

15 Dieser Erfolg des Internet und der Intranets wurde stark durch die Einführung des World Wide Web (WWW) gefördert, das von Internet-Benutzern einfach bedient werden kann und erlaubt, Informationen unterschiedlichster Art und von unterschiedlichsten Orten der gesamten Welt abzurufen. Der WWW-Dienst des Internets  
20 beruht auf dem Client-Server-Prinzip. Die Kommunikation erfolgt zwischen einem Web-Server, der auch als WWW-Server bezeichnet wird und der Informationen bereitstellt, und einem Client, der die Informationen anzeigt. Die Informationen sind auf dem Web-Server in Seiten gespeichert, wobei die Daten in den Seiten z. B. im sogenannten HTML-Format (**Hypertext Markup Language**) abgespeichert sind. Es sind auch andere Formate gebräuchlich, wie z. B. XML. Diese Formate sind Derivate des grundlegenden SGML-Formates (**Standard General Markup Language**). Diese Formate werden als Markup-Sprachen bezeichnet, da mit ihnen die Aufmachung  
30 beschrieben und festgelegt werden kann.

Ein entsprechendes Dokument besteht aus normalem Text, bei dem Steueranweisungen, sogenannte „Tags“ in den Text eingefügt sind. Diese Tags beeinflussen unter anderem das Layout, das  
35 später im Betrachtungsprogramm, dem Browser, am Client angezeigt wird. So gibt es z.B. Tags um Überschriften zu erzeugen,

oder Tags, die das Schriftbild verändern können. Die Tags werden immer in „< ...>“ eingeschlossen.

Die Übertragung der Informationen vom Web-Server zum Client er-  
folgt gemäß dem HTTP-Protokoll, wobei die an den Client über-  
tragene Information vom dort installierten Browser gelesen und  
die einzelnen Tags interpretiert werden, so daß die Information  
in der vorbestimmten Art und Weise am Bildschirm des Client  
dargestellt werden.

10

Da mit dem HTML-Format im wesentlichen nur statische Bilder und  
Texte dargestellt werden können, ist der WWW-Dienst zur Einbin-  
dung von Multimedia-Elementen, Animationen oder Programmen  
durch JAVA ergänzt worden. JAVA ist eine von SUN Microsystems  
15 entwickelte, objektorientierte Programmiersprache für Web-  
Anwendungen. JAVA unterstützt Text-, Hypertext-, Grafik-, Au-  
dio- und Animationsfunktionen.

Die in JAVA programmierten Programmpakete können entweder als  
20 sogenannte JAVA Applets vom Web-Server auf den Client geladen  
werden oder am Client als sogenannte JAVA-Applikationen vorin-  
stalliert sein. Die JAVA-Applets sind am Server separat von den  
dazugehörigen HTML-Seiten gespeichert. Eine Abwandlung von JA-  
VA, die als JAVAscript bezeichnet wird, erlaubt das Einfügen  
von Programmpaketen in der HTML-Seite.

Um über das Internet Drucker und Drucksysteme ansteuern zu kön-  
nen, hat man JAVA-Applikationen entwickelt, mit welchen man so-  
wohl Verwaltungsfunktionen an den Druckern und Drucksystemen  
30 als auch das Absenden von Druckaufträgen an die einzelnen Druk-  
ker bzw. Drucksysteme ausführen kann. Diese JAVA-Programme  
stellen einen wesentlichen Fortschritt gegenüber dem bisherigen  
Zustand dar, da es dem Benutzer eines Client-Rechners ermög-  
licht, über das Internet mehrere Drucker und Drucksysteme  
35 plattformunabhängig anzusteuern. Die Art der Drucksysteme, das  
heißt, die Schnittstellentypen, die von einem Client angesteu-  
ert werden können, sind durch die jeweilige JAVA-Applikation

festgelegt, wobei für jeden Schnittstellentyp ein separates  
Programmelement vorgesehen ist. Dies bedeutet, daß je mehr un-  
terschiedliche Schnittstellen mit einer solchen JAVA-  
Applikation angesteuert werden sollen, desto umfangreicher muß  
5 diese JAVA-Applikation sein. Entsprechendes gilt für den Funk-  
tionsumfang, der bei einer solchen JAVA-Applikation angesteuert  
werden soll. Dies hat zur Folge, daß diese JAVA-Applikationen  
umfangreiche Programme mit z.B. einen Datenumfang von 9 MB  
sind. Sie beanspruchen somit am Client erhebliche Rechenlei-  
10 stung und Speicherplatz und sind deshalb nur für Anwender  
zweckmäßig, die regelmäßig unterschiedliche Drucker und Druck-  
systeme ansteuern wollen.

Es ist auch bekannt, am Server ausführbare Programme vorzuse-  
15 hen, die vom Client aufgerufen werden können. Diese Programme  
sind entweder als vollständig compilierte Programme oder als  
von einem Interpreter interpretierbarer Programmcode am Server  
hinterlegt. Für einen derartigen interpretierbaren Programmcode  
wird für Internetanwendungen häufig die Programmiersprache Perl  
20 verwendet. Sie ist insbesondere für am Server auszuführende Da-  
tenbankanwendungen geeignet. Mit Perl können jedoch auch vom  
Server zum Client ladbare Seiten erzeugt werden, wobei Perl Be-  
fehle vorsieht, mit welchen die Aufmachung entsprechender Sei-  
ten aufgebaut werden kann. Hierbei kann auch vom Perl-  
25 Programmcode ein HTML-Programmcode generiert werden. Beim Auf-  
ruf eines derartigen Programmes am Server wird der vollständige  
Programmcode am Server interpretiert und ausgeführt. Auf Perl  
basierende Programme zum Ansteuern von Druckern, Drucksystemen  
und entsprechenden Vor- und Nachbearbeitungsgeräten sind dem  
30 Erfinder der vorliegenden Erfindung nicht bekannt.

Aus dem Druckerbuch der Océ Printing Systems GmbH, ISBN 3-00-  
001019-X, Ausgabe 3d, Oktober 1998 geht auf den Seiten 12-2 bis  
12-8 ein Print-Server für professionelle, Internet-unabhängige  
35 Anwendungen hervor. Dieser Print-Server wird als PRISMApro® be-  
zeichnet. Ein solcher Print-Server besteht aus einem leistungs-  
fähigen Personalcomputer, mit einer entsprechenden Software, um

einen oder mehrere Drucker, insbesondere einen Schnell- oder Hochleistungsdrucker ansteuern zu können. Ein solcher Print-Server ist in ein Netzwerk eingebunden und setzt die einkommenden Datenströme in entsprechende Druckdaten um. Je nach Ausführungsform kann der Print-Server Daten in den Formaten AFDPS, Line/SF, PostScript, TIFF, PDF, LCDS, Line/O und PCL verarbeiten. Derartige Print-Server werden insbesondere bei Datenbankanwendungen eingesetzt, wobei aus einer Datenbank eine große Datenmenge mit variablen Daten gedruckt wird.

10

Ferner ist im Druckerbuch der Océ Printing Systems GmbH, ISBN 3-00-001019-X Ausgabe 3d, Oktober 1998 auf den Seiten 14-2 bis 14-12 das Druck-Produktionskontrollsystem Océ Domain® beschrieben. Bei diesem System werden ein oder mehrere Drucker mit Vor- und Nachverarbeitungsgeräten mittels eines Netzwerkes verbunden. Das System steuert und überwacht die Druckproduktion. Es ist eine netzwerkbasierende Client-Server-Lösung mit einer Datenbank, die auf einem zentralen Datenbankserver installiert ist. Alle im Produktionsprozeß erfaßten Maschinen- und Betriebsdaten werden in dieser Datenbank gesammelt und für spätere Auswertungen den verschiedenen Clients bereitgestellt oder können in kundenspezifische Anwendungen exportiert werden. Mit Océ Domain® werden Schnittstellen zu dem im Netzwerk angegeschlossenen Hochleistungsdruckern und den verschiedenen Hard- und Softwarekomponenten realisiert. Es werden die Industrie-standards DMI (Desktop Management Interface), LMO (Large Mailroom Operations) und ODBC (Open Data Base Connectivity) unterstützt. Océ Domain® ist ein Hochleistungssteuer- und Kontrollsysteem, das insbesondere bei Druckzentren zur Anwendung kommt, bei welchen Hochleistungsdrucker mit Geräten zur Vorverarbeitung und zur Nachverarbeitung gekoppelt sind.

Der Erfindung liegt die Aufgabe zugrunde, eine einfache technische Lösung zur Ansteuerung unterschiedlicher Drucker und Drucksysteme zu finden.

Die Aufgabe wird durch ein Netzwerk mit den Merkmalen des Anspruchs 1, einen Interpreter für ein derartiges Netzwerk mit den Merkmalen des Anspruchs 9 und durch ein Verfahren zum Betreiben eines Netzwerkes mit den Merkmalen des Anspruchs 15 ge-  
5 löst. Vorteilhafte Ausgestaltungen der Erfindung sind in den Unteransprüchen angegeben.

Das Netzwerk gemäß der Erfindung ist ein Client-Server-Netzwerk, das zumindest einen Client mit zumindest einem Server  
10 verbindet, wobei am Server gespeicherte Dateien vom Client durch Mitteilen einer Dateiadresse abgerufen werden können, wo-  
durch eine entsprechende Datei an den Server übertragen wird.  
Diese vom Server an den Client übertragbaren Dateien enthalten Sprachelemente, die am Client ausgeführt werden.

15 Die Erfindung zeichnet sich dadurch aus, daß am Server ein Interpreter vorgesehen ist, der weitere, am Server ausführbare Sprachelemente, die in den am Server gespeicherten und vom Client abrufbaren Dateien enthalten sind, interpretieren und aus-  
20 führen kann.

Mit der Erfindung werden somit erstmals am Server Dateien vor-  
gesehen, die sowohl am Server ausführbare Sprachelemente als auch am Client ausführbare Sprachelemente enthalten. Hierdurch werden wesentliche Vorteile erzielt, da es nicht mehr notwendig ist, wie es z. B. von am Server ausführbaren Perl-Programmen bekannt ist, am Client ausführbare Sprachelemente durch Sprachelemente einer anderen Sprache zu generieren, was mit erheblichem Programmieraufwand verbunden ist.

30 Entsprechen die am Client ausführbaren Sprachelemente einer Markup-language (wie z. B. SGML, HTML, XML) können die Dateien mit den meisten gebräuchlichen Editoren und Textverarbeitungsprogrammen editiert werden. Zur Erstellung einfacher Markup-  
35 Dateien sind oftmals keine Kenntnisse über die Markup-Sprache notwendig, da die entsprechende Syntax automatisch vom Editor bzw. vom Textverarbeitungsprogramm eingefügt wird. Mit der Er-

findung können somit komplexe Anwendungen mit einfach zu gebrauchenden Textverarbeitungsprogrammen erstellt werden. Er muß lediglich in die ihm geläufigen vom Server aufrufbaren Dateien die am Server ausführbaren Sprachelemente hinzufügen. Die Kombination von am Client ausführbaren als auch am Server ausführbaren Sprachelementen in einer Datei stellt somit eine wesentliche Erleichterung für den Benutzer dar, der die Dateien am Server einrichtet, da er mit seinen ihm vertrauten Mitteln diese Dateien erstellen kann und die am Client ausführbaren Befehle nicht in einer anderen Computersprache codieren muß, bei deren Ausführung sie künstlich erzeugt werden.

Durch das Vorsehen eines Interpreters, der die vom Client abgerufenen Dateien auf darin enthaltene Sprachelemente untersucht und gegebenenfalls diese Sprachelemente am Server interpretiert und ausführt, kann durch die Übertragung lediglich einer einzigen Adresse vom Client zum Server die Ausführung eines an sich beliebigen Programmes am Server initiiert werden. Es wird somit eine Möglichkeit geschaffen, mit minimalem Programmieraufwand am Client, und minimalstem Datenvolumen, das zwischen dem Server und dem Client übertragen werden muß, beliebige, am Server vorab in entsprechenden Dateien hinterlegte Programme, aufrufen zu können.

Diese Programme können von mehreren Clients benutzt werden, so daß lediglich an einer einzigen Stelle im Netzwerk, am Server, ein entsprechendes Programm Paket hinterlegt werden muß.

Die mit der Erfindung erzielten Vorteile werden besonders deutlich, wenn die Datenübertragung im Netzwerk auf einer standardisierten Datenübertragung basiert, wie z.B. der gemäß dem HTTP-Protokoll, das bei den Intra- und Internets angewandt wird. Bei Verwendung eines derartigen Standards wird am Client lediglich ein herkömmlicher Browser ohne weitere zusätzliche Programme benötigt, um die entsprechenden Funktionen anzusteuern. Bisher wurden zum Ansteuern derartiger Funktionen, wie es eingangs erläutert worden ist, aufwendige Programme z. B. in

JAVA auf den Clients geladen, wodurch lediglich an den Clients die gewünschten Funktionen angesteuert werden können, auf welche die entsprechenden Programme installiert oder geladen worden sind. Bei dem erfindungsgemäßen Netzwerk können hingegen  
5 von jedem Client, an dem ein Browser vorgesehen ist, die entsprechenden Programme am Server aufgerufen werden, ohne daß zusätzliche Programme auf den Client geladen werden müssen. Man benötigt lediglich eine Zugangsberechtigung zu dem jeweiligen Server.

10 Ein wesentlicher funktioneller Vorteil der Erfindung gegenüber den herkömmlichen Systemen liegt darin, daß es gemäß der Erfindung möglich ist, daß in den am Server abgespeicherten Dateien  
15 Programmelemente enthalten sind, die den Client automatisch zu einem weiteren Server vermitteln. Diese Vermittlung erfolgt z.B. einfach durch Übertragen der Adresse des weiteren Servers vom ursprünglichen Server zum Client, die dann am Client ausgeführt wird, das heißt zu weiteren Server geschickt wird, wo-  
durch eine Verbindung zwischen dem weiteren Server und dem Cli-  
20 ent aufgebaut wird.

Diese Vermittlungsfunktion kann auch als automatische Vermittlungsfunktion ausgestaltet sein, wobei am Server eine Datei angelegt ist, die die Dienste anderer Server beinhaltet, so daß,  
25 wenn von einem Client ein bestimmter Dienst an diesem einen Server abgefragt wird, und dieser Server den Dienst nicht erfüllen kann, den Client automatisch zu einem weiteren Server vermittelt, der den Dienst erledigen kann. Für den Client stellt sich diese Funktion so dar, als ob sein Auftrag zwischen  
30 den Servern ausgehandelt wird, um denjenigen Server festzustellen, der ihn erfüllen kann. Man bezeichnet diese Funktion der automatischen Weitervermittlung deshalb auch als „Trading“.

Das Trading umfaßt auch die Möglichkeit, daß der Server als eine Art Durchgangs- bzw. Relaisstation fungiert, wobei er eine Verbindung zwischen dem Client und einer weiteren Arbeitsstation wie z.B. einem weiteren Server einer Druckstation oder der-

gleichen schaltet, bei der der Server zwischen der Arbeitsstation und dem Client geschaltet ist und die Daten entsprechend weiterleitet. Ist der Server in einem solchen Fall mit einem Gateway versehen, so kann zwischen der weiteren Arbeitsstation 5 und dem Server und dem Client die Datenübertragung auf einem unterschiedlichen System bzw. unterschiedlichen Protokoll beruhen, wobei die Daten vom Server entsprechend übersetzt werden, damit eine reibungslose Datenübertragung möglich ist.

10

Ein Drucksystem mit einem solchen Client-Server-Netzwerk erlaubt bei der Hinterlegung entsprechender Dateien am Server jedem Client, der auf diesen Server zugreifen kann, die mittels des Servers ansteuerbaren Druckeinrichtungen und Vor- und Nachbearbeitungseinrichtungen anzusprechen, ohne daß hierfür am Client ein spezielle Software installiert sein muß. Die einzelnen Druckaufträge können auch von einem Server an einen anderen Server weitervermittelt werden, um z.B. auf einem für diesen speziellen Druckauftrag besonders geeigneten Drucker ausgedruckt zu werden. Man kann den Server mit Gateways zum Umsetzen der Daten in spezielle Druckprotokolle oder auch andere Arten von Übertragungsprotokollen versehen, so daß mittels eines einzigen Servers auf unterschiedlichste Art und Weise an das Netzwerk angekoppelte Drucker angesteuert werden können.

2

Eine aus mindestens einem Drucker, mit evtl. einem Vor- und Nachbearbeitungsgerät bestehende Druckstation kann einfach an ein Netzwerk angekoppelt werden, indem die Geräte der Druckstation (Drucker, Vor- und Nachbearbeitungsgeräte) mit einem Server verbunden werden, der mit dem erfindungsgemäßen Interpreter versehen ist, und an dem entsprechende Programme zur Ansteuerung der einzelnen Geräte der Druckstation in Dateien abgespeichert sind, die vom Client abgerufen werden können. Die Geräte einer solchen Druckstation können dann über das Netzwerk von jedem beliebigen Client angesteuert werden, sofern eine entsprechende Zugangsberechtigung zum Server vorliegt. Die Erfindung schafft somit eine Möglichkeit, mit der man lediglich

durch Hinzufügen eines Servers an eine bestehende Druckstation einen Netzzugang zur Druckstation erstellt, der mit äußerst einfachen technischen Mitteln angesprochen werden kann.

5 Das erfindungsgemäße Netzwerk ist zur Ansteuerung von Druckern und entsprechenden Vor- und Nachbearbeitungsgeräten ausgebildet. Es kann jedoch auch zur Ansteuerung beliebiger anderer Geräte verwendet werden, da das Grundprinzip der vorliegenden Erfindung, das Vorsehen eines Interpreters an einem Netz-Server,  
10 der vom Client aufgerufene Dateien interpretiert, auf weitere, beliebige Netzwerkanwendungen übertragen werden kann.

Nachfolgend wird die Erfindung beispielhaft anhand der beigefügten Zeichnungen näher erläutert. In denen zeigen:

15 Fig. 1 schematisch ein erfindungsgemäßes Netzwerk in einem Blockschaltbild,

20 Fig. 2 eine Schnittstellenabfrage in einem Flußdiagramm, das den Programmablauf des der als Anhang beigefügten Computerprogramms zeigt, und

Fig. 3 ein Tradingverfahren im Flußdiagramm.

25 Fig. 1 zeigt ein Ausführungsbeispiel eines erfindungsgemäßen Netzwerkes. Das Netzwerk weist einen ersten Web-Server 1 einen zweiten Web-Server 2 und einen ersten und zweiten Client 3, 4 auf. Der erste und zweite Web-Server und der erste Client sind mittels eines LANs über Datenleitungen 5, 6 miteinander verbunden, wobei die Datenleitung 5 den ersten Web-Server mit dem ersten Client und die Datenleitung 6 den ersten Web-Server 1 mit dem zweiten Web-Server 2 verbindet. Auf dem LAN wird ein Intranet betrieben.

35 Zwischen dem ersten Web-Server 1 und dem zweiten Client 4 besteht keine feste physikalische Datenverbindung. Bei Bedarf wird z.B. über das Telefonnetz eine entsprechende Datenverbin-

dung 7 aufgebaut wird, wobei das Internet als Übertragungsmedium verwendet wird.

Der erste Client 3 ist z.B. ein über das Intranet mit dem ersten Web-Server 1 verbundener Bürocomputer eines Anwenders des erfindungsgemäßen Netzwerkes, wohingegen der Client 2 ein zu Hause beim Anwender stehender Personalcomputer ist, mit dem sich der Anwender mittels eines Modems im Internet einwählen kann und eine Verbindung zum ersten Web-Server 1 herstellen kann. Da die Datenverbindung 7 nicht permanent vorliegt, ist sie in Fig. 1 gestrichelt eingezeichnet.

An beiden Clients 3, 4 ist jeweils ein Browser installiert, so daß die beiden Clients 3, 4 mit dem Web-Server 1 mit dem vom Internet bekannten Diensten kommunizieren können. Diese Dienste sind z.B. Telnet, das eine Terminal-Simulation ermöglicht, oder FTP, mit dem Dateien übertragen werden können. Bei dem vorliegenden Ausführungsbeispiel wird der Dienst des World Wide Web (WWW) verwendet, der bei Bedarf automatisch auf die weiteren Dienste, wie z.B. FTP, News, Telnet, Gopher, E-mail, usw. des Internets zurückgreift. An dem ersten Web-Server 1 sind zwei Geräte 8, 9 angeschlossen, wobei das Gerät 8 mittels einer seriellen Leitung (RS 232/V24) mit dem Web-Server 1 verbunden ist und zwischen dem Gerät 9 und dem Server 1 eine Verbindung 11 nach dem SNMP-Protokoll (Simple Network Management Protokoll) installiert ist. Das SNMP-Protokoll wird insbesondere zum Ansteuern von an einem Netzwerk angeschlossenen Geräten verwendet. Am Web-Server 1 ist ein Gateway 12 vorgesehen, das die über die SNMP-Verbindung 11 einkommenden Daten vom SNMP-Protokoll in das vom Web-Server 1 auf den Datenverbindungen 5 bis 7 angewandte Internet-Protokoll umsetzt bzw. in umgekehrter Richtung die Daten gemäß dem Internet-Protokollen auf das SNMP-Protokoll umsetzt, wenn sie vom Web-Server 1 zum Steuergerät 9 übertragen werden. Ein Gateway im Sinne der Erfindung sind alle Hard- und Softwarekomponenten am Web-Server, die eine Verbindung zu einem sich vom Internet unterscheidenden Kommunikationssystem bzw. Netzwerkes herstellen. Am Web-Server 1 können

mehrere weitere Gateways zu anderen Kommunikationssystemen vorgesehen sein, die z.B. auf dem DMI-Standard (Desktop Management Interface), dem LP-Standard (Line Printer), SLP (Service Location Protocol), dem IPP-Standard (Internet Printing Protokoll) beruhen. Es sind Gateways zum PJMweb möglich. PJMweb ist eine web-basierender Druckclient, der von Océ entwickelt worden ist. Ein Gateway kann auch eine Terminal-Emulation erzeugen. Insbesondere können Gateways vorgesehen werden, die zur Ansteuerung spezieller Druckerkommunikationssysteme ausgebildet sind.

10

Da über die Gateways eine Verbindung zu unterschiedlichen Netzwerken hergestellt werden kann, können an einen mit einem Gateway versehenen Server grundsätzlich beliebig viele Geräte angeschlossen bzw. vom Server angesteuert werden.

15

Der zweite Web-Server 2 ist mit einem weiteren Gerät 13 über eine Leitung 14 verbunden. Hierzu ist am zweiten Server 2 ein speziell für das Gerät 13 ausgebildetes Kommunikationsprogramm 15 installiert. Dieses Kommunikationsprogramm 15 ist ein compiliertes Programm für ein sogenanntes Web-Based-Management. Derartige Programme sind jeweils für eine spezielle Anwendung ausgerichtet, wie z.B. der Ansteuerung des Gerätes 13, wodurch mit diesem Programm lediglich eine einzelne Verbindung zu dem Gerät 13 hergestellt und lediglich ein bestimmter Gerätetyp angesteuert werden kann.

Der erste und zweite Web-Server 1, 2 weisen neben anderen Dienstprogrammen ein WWW-Dienstprogramm 16 auf, das die aus dem Stand der Technik bekannten Funktionen aufweist, das heißt, daß es bei Empfang eines URL aus einem Datenspeicher 17 des Webserver 1, 2 die entsprechenden Seiten ausliest und an den Client 3, 4 schickt, der die URL abgeschickt hat. Erfahrungsgemäß ist am Server 1, 2 ein Interpreter 18 vorgesehen, mit welchem die im Speicher 17 abgelegten Seiten vor dem Versenden zu einem Client 3, 4 interpretiert werden, das heißt, daß in den Seiten enthaltene Sprachelemente vom Interpreter 18 ausgeführt werden. Die einzelnen Seiten sind im HTML-Format abgespeichert, das

heißt, daß sie standardisierte Sprachelemente aufweisen, die vom Browser des Clients 3, 4 ausgeführt werden können. Die vom Interpreter 18 ausführbaren Sprachelemente sind unabhängig von dem vom Browser ausführbaren Sprachelementen. Sie können auch 5 als Ergänzung zum HTML-Format betrachtet werden, weshalb das Format der abgespeicherten Seiten auch als erweitertes HTML-Format bezeichnet werden kann.

Im Anhang ist ein Programmcode eines Programmbeispiels angegeben, das unten näher erläutert wird. Zunächst wird der Befehlsumfang des Interpreters kurz erläutert. Der Interpreter umfaßt, wie die meisten Computersprachen, Befehle zur Stringverarbeitung, Systembefehle und Strukturbefehle. Der Interpreter kann je nach Bedarf mit weiteren Gruppen von Befehlen versehen 15 werden. Zusätzlich zu den vorgenannten Gruppen von Befehlen weist der Interpreter einen Befehl auf, mit welchem neue Befehle des Interpreters erzeugt werden können. Dieser Befehl ist „addfunction“, mit dem ein beliebiger Befehl (Funktion) aus beliebigen auf dem Server zur Verfügung stehenden Befehlen (MaschinenSprache, Betriebssysteme oder sonstige Hochsprache) aufgebaut werden kann. „addfunction“ wird nicht in den HTML-Seiten 20 verwendet, sondern kann vom Benutzer in einer Programmentwicklungsumgebung aufgerufen werden.

Wichtige Stringverarbeitungsbefehle sind „userDefineString“, „userGetCGIString“, „userPreReplaceString“, „userPostReplaceString“ und „userCompose“. Mit „userDefineString“ können Stringvariablen definiert werden. Mit „userGetCGIString“ kann ein am Client eingegebener String eingelesen werden. Die Befehle 30 „userPreReplaceString“ und „userPostReplaceString“ dienen zum Anordnen eines Strings an eine vorbestimmte Stelle der Seite, wobei mit dem Befehl „userPreReplaceString“ der String unmittelbar nach Ausführung dieses Befehls an der entsprechenden Stelle angeordnet wird, wohingegen beim „userPostReplaceString“ 35 der String erst nach Abarbeitung aller Sprachelemente der Seite an der vorbestimmten Stelle angeordnet wird. Der durch den Befehl „userPreReplaceString“ dargestellte String kann somit

durch die weiteren vom Interpreter auszuführenden Sprachelemente noch verändert werden, wohingegen der durch den Befehl „userPostReplaceString“ angeordnete String am Ende plaziert wird, wodurch eine Veränderung des Strings nicht mehr möglich ist.

Einer der mächtigsten Systembefehle ist „userSystem (...)“, wo bei in die Klammer ein Befehl des Betriebssystems des Servers oder eines weiteren auf dem Server installierten Programmes 10 eingegeben werden kann. Hierdurch können mit dem Interpreter am Server eingerichtete Befehle und Programme aufgerufen werden. Dieser Befehl wird aus einer HTML-Seite aufgerufen.

Eine Untergruppe der Systembefehle sind Steuerungsbefehle, die 15 z.B. zum Ansteuern eines bestimmten Druckers dienen. Diese Befehle entsprechen in der Regel dem jeweiligen Druckersystembefehlen, die als Befehl des Interpreters jeweils umgesetzt werden und durch weitere, höhere Steuerungsbefehle ergänzt werden können.

20 Die Strukturbefehle dienen zum Erstellen einer Programmstruktur mit Verzweigungen, Schleifen und dergleichen. Entsprechende Strukturbefehle sind z.B. „userFor“, „userGoSub“ oder „userIf“.

18 Entsprechend der Syntax von HTML werden auch die vom Interpreter ausführbaren Befehle in „<...>“ gesetzt.

Bei den Befehlen des Interpreters 18 unterscheidet man zwei Klassen, nämlich eine mit Befehlen, die nicht direkt vom Client 30 aufgerufen werden können, und eine weitere Klasse, deren Befehle direkt vom Client aufgerufen werden können. Von den oben beispielhaft angegebenen Befehle können diejenigen nicht vom Client aufgerufen werden, die mit „user“ beginnen. Diese Unterteilung in zwei Klassen der Befehle dient der Sicherheit, denn 35 wenn ein Anwender von einem beliebigen Client einen „userSystem“ Befehl auf dem Server erzeugen könnte, könnte er den Server nach Belieben manipulieren. Zur Vermeidung einer solchen

Manipulation ist der Speicher 17 in einen freien Speicherbereich 17a und in einen gesperrten Speicherbereich 17b unterteilt. Im freien Speicherbereich können vom Client Seiten aufgerufen werden und diese ggf. mit Parametern versorgt werden,

5 wohingegen der gesperrte Speicherbereich 17b Seiten enthält, die nicht direkt vom Client Parameter erhalten oder von ihm aufgerufen werden können. Diese Seiten werden lediglich vom Interpreter aufgerufen. Der Interpreter kontrolliert hierbei die Parameterübergabe und die Programmausführung. Der Interpreter

10 18 ist derart ausgebildet, daß sicherheitsrelevante Befehle nur ausgeführt werden, wenn sie im gesperrten Speicherbereich 17b abgespeichert sind. Ein Anwender an einem der Clients kann lediglich den Inhalt des freien Speicherbereiches 17a lesen und direkt adressieren. Die im gesperrten Speicherbereich 17b abgelegten Seiten können nur indirekt über den Interpreter adressiert werden.

Der Administrator des Servers kann im gesperrten Speicherbereich 17b des Servers die zum Ansteuern der Geräte 8, 9 und 13 notwendigen Steuerprogramme hinterlegen, die in der Regel sicherheitsrelevant sind.

Nachfolgend wird der im Anhang angegebene Programmcode näher erläutert, dessen Programmablauf in Fig. 2 in einem Flußdiagramm dargestellt ist. Im Schritt S1 „ini“ wird eine Initialisierung durchgeführt, wobei bestimmte Werte und Strings für jeweils abzufragende Server gesetzt bzw. definiert werden.

Mit dem Schritt S2 wird das SNMP-Gateway 12 aufgerufen, wobei

30 durch den Wert „1“ angegeben wird, daß von diesem Gateway etwas eingelesen werden soll, wobei das Ergebnis in „SNMPVALUE“ gespeichert wird. Hierbei wird in Abhängigkeit von einer Benutzereingabe eine Adresse eines Datenbankeintrages (Managed Object einer MIB) abgefragt. Das Ergebnis dieser Abfrage stellt

35 den Rechntertyp dar, der über die SNMP-Schnittstelle angesprochen wird.

Anhand des in der Variablen „SNMPVALUE“ abgespeicherten Wertes wird im nächsten Schritt S3 geprüft, ob das Betriebssystem des Über das SNMP-Gateway angesprochenen Rechners Windows ist.

Falls das Betriebssystem Windows ist, verzweigt der Pro-

grammablauf auf die Abfrage S4, bei der erneut ein SNMP-Wert eingelesen wird und anhand dieses Wertes geprüft wird, ob an dem Rechner ein vorbestimmtes, unter Windows lauffähiges Druck-  
system („Imagestream“) eingerichtet ist. Falls die Abfrage S4 das Vorhandensein eines derartigen Drucksystems ergibt, wird dies im Schritt S5 abgespeichert.

Nach dem Speichervorgang im Schritt S5 bzw. falls eine der Ab-  
fragen S3 oder S4 verneint worden ist, geht der Programmablauf auf den Schritt S6 über, mit dem geprüft wird, ob das Betriebs-

system des über das SNMP-Gateway angesprochenen Rechners ein Unix-Betriebssystem ist. Ergibt die Abfrage, daß das Betriebs-  
system ein Unix-Betriebssystem ist, wird der Programmablauf auf den Schritt S7 verzweigt, mit dem wiederum ein SNMP-Wert einge-  
lesen wird, anhand dessen geprüft wird, ob ein Unix-  
Betriebssystem (PJM bzw. Prisma) an dem Rechner eingerichtet ist. Ergibt die Abfrage ein derartiges Drucksystem, so wird dies im Schritt S8 abgespeichert.

Nach dem Speichervorgang des Schrittes S8 oder wenn eine der beiden Abfragen der Schritte S6 und S7 negativ ausgefallen ist, geht der Programmablauf auf einen Schritt S9 über, mit dem ge-  
prüft wird, ob der über das SNMP-Gateway angesprochene Rechner überhaupt ein mittels des SNMP-Protokolls ansteuerbarer Rechner ist. Ergibt die Abfrage, daß der Rechner mittels des SNMP-

Protokoll grundsätzlich ansteuerbar ist, so wird im Schritt S10 eine PING-Abfrage aufgerufen und im Schritt S11 das Ergebnis der PING-Abfrage geprüft, ob der Rechner in Betrieb (on-line) ist. Ergibt diese Abfrage S11, daß der Rechner in Betrieb ist (on-line), so wird dies im Schritt S12 gespeichert, wohingegen, wenn der Rechner nicht in Betrieb (off-line) ist, dann entspricht dies der default-Einstellung, so daß keine zusätzliche Abspeicherung notwendig ist.

Die PING-Abfrage ist ein nicht nur im Internet vorhandener Programmabaustein, der an vielen Servern eingerichtet ist. Er wird deshalb mit der bereits oben beschriebenen Funktion „userSystem“ aufgerufen oder als eigenständiger Befehl in den Interpreter integriert. Eine erneute Programmierung dieses Programmabausteins erübriggt sich somit.

Nach dem Speichervorgang des Schrittes S12 oder falls die Abfrage im Schritt S9 ergab, daß der Rechner ein SNMP-Rechner ist oder falls die Abfrage im Schritt S11 ergab, daß der Rechner nicht in Betrieb ist, geht der Programmablauf auf den Schritt S13 über, mit dem ein das Drucksystem darstellendes Bild aufgenommen wird. Dies erfolgt mit dem Befehl „userCompose“, mit dem die durch die obigen Abfragen und Speichervorgänge belegte Variable „SNMPVALUE“ ausgewertet wird (siehe Zeile unterhalb des Befehles „userCompose“), wobei das Ergebnis „exklusiv“ ist. Dies ist in der darauffolgenden Zeile durch den Wert „false“ vorgegeben. Dies bedeutet, daß nur ein einziges Bild in die Variable „REPLACEMENT“ eingelesen werden kann. Je nach dem obigen Ergebnis, wird ein Bild für ein, ein Imagestream-Drucksystem (imagestream.gif), ein Prisma-Drucksystem (Prisma.gif) oder ein Bild für eine erfolgreiche PING-Abfrage (ping.gif) oder ein Bild für den nicht-Betriebszustand des Rechners (offline.gif) oder ein Bild für einen mittels des SNMP-Protokoll ansprechbaren Rechners (SNMP.gif) eingelesen.

In den nachfolgenden Schritten S14 und S15 wird wiederum mit-tels des Befehls „userCompose“ eine Hyperlink-Start-Variabile („SnmpHLStart“) bzw. eine Hyperlink-Ende-Variabile („SnmpHLEnd“) mit den zum Erzeugen eines Hyperlinks notwendigen Strings belegt. Ein Hyperlink ist ein automatisierter Aufruf einer Datei des Servers, wobei ein Client die entsprechende URL an den die Datei aufweisenden Server sendet.

35

In Figur 2 ist gestrichelt ein Programmmpfeil vom Schritt S15 zum Schritt S2 eingezeichnet, der den Programmabschnitt zwi-

990503 DE

schen den Schritten S2 und S15 zu einer Schleife schließt. Mit diesem Programmpeil wird ausgedrückt, daß dieser Programmabschnitt mehrfach zum Abfragen mehrerer über das SNMP-Gateway erreichbare Rechner ausgeführt wird. Die Abfragen der einzelnen  
5 Rechner werden parallel ausgeführt. Dies ist möglich, da das SNMP-Gateway multitasking-fähig ist.

Im nachfolgenden Schritt S16 werden die den Hyperlink darstellenden Strings mittels des Befehls „userPostReplaceString“ in  
10 die Datei bzw. HTML-Seite an eine vorbestimmte Stelle einge-fügt.

Mit dem oben beschriebenen Programm wird somit abgefragt, ob mit dem SNMP-Gateway ein oder mehrere bestimmte Rechner ange-  
15 sprochen werden können und ob an einem der ansprechbaren Rechner ein bestimmtes Drucksystem eingerichtet ist. Ein entspre-chendes Bild wird dann eingelesen und ein auf den Rechner ver-weisender Link generiert und in der HTML-Seite abgelegt. Wird nun die HTML-Seite zum Client übertragen, so kann entweder der  
20 Link manuell oder automatisch ausgeführt werden, womit eine Verbindung vom Client, der die Abfrage gestartet hat, direkt zum Rechner mit dem abgefragten Drucksystem hergestellt wird.

Der Client wurde somit von einem Server zu einem weiteren Ser-ver vermittelt, wobei der erste Server für den Client ein Drucksystem gesucht hat, mit dem vom Client aus dann direkt entsprechende Druckaufträge ausgeführt werden können. Die Such- und Vermittlungsprozedur ist vollständig am Server ausgeführt worden, das zeigt, daß die notwendige „Intelligenz“ lediglich  
30 am Server vorgesehen sein muß und vom Client mit einem herkömm-lichen Browser abgefragt und bedient werden kann.

Das obige Programm ist lediglich ein stark vereinfachtes und verkürztes Beispiel für eine Abfrage von erreichbaren Drucksy-  
35 stemen und automatischer bzw. halbautomatischer Vermittlung an ein gewünschtes Drucksystem. Dieses Beispiel soll lediglich an-

deuten, welche Möglichkeiten durch das erfindungsgemäße Vorsehen eines Interpreters am Server geschaffen werden.

In Fig. 3 ist ein Flußdiagramm dargestellt, mit welchem gezeigt wird, wie ein Druckauftrag mit dem erfindungsgemäßen Netzwerk abgearbeitet werden kann. Ein Anwender gibt in einem Schritt S18 seinen Druckauftrag ein und ergänzt ihn durch bestimmte Wünsche an die Qualität der Druckerzeugnisses (farbig, Papierart, usw.) und notwendigen Druckmerkmalen (Anzahl der Kopien, Format, usw.).

Dieser Druckauftrag wird im Schritt S19 vom Client 3, 4 zum Server 1, 2 übertragen. Am Server wird eine Datenbasis, die die relevanten Daten für Verbindungen zu Druckern bzw. weiteren Servern mit angeschlossenen Druckern bzw. entsprechenden Vor- und Nachbearbeitungsgeräten enthält, aktualisiert S20. Die Aktualisierung S20 erfolgt, indem eines oder mehrere Gateways nach entsprechenden Drucksystemen abgefragt werden. Die hierbei ermittelten Daten werden in der im Server abgelegten lokalen Datenbasis eingetragen.

Diese aktualisierte Datenbasis wird nach den vom Anwender eingegebenen Parametern (Druckauftrag, Wünsche und Voraussetzungen) im Schritt S21 ausgewertet.

Anschließend wird mit einer Abfrage S22 geprüft, ob bei der Auswertung eine für den Druckauftrag geeignete Druckeinrichtung gefunden worden ist. Falls keine geeignete Druckeinrichtung gefunden worden ist, zweigt der Programmablauf auf den Schritt S23 ab, mit dem an den Client eine Nachricht gesandt wird, daß der Druckauftrag nicht ausführbar ist.

Ergibt die Abfrage im Schritt S22 hingegen, daß eine geeignete Druckeinrichtung vorhanden ist, wird mit einer weiteren Abfrage S24 geprüft, ob der Druckauftrag vom Server ausgeführt werden kann. Falls das Ergebnis dieser Abfrage nein ist, wird, wie es anhand des Programmbeispiels aus Fig. 2 dargestellt worden

ist, der Client zu einem weiteren Server vermittelt S25, der den Druckauftrag ausführen kann. Hierbei wird vorzugsweise der Druckauftrag zusammen mit den Parametern, (Wünsche und Voraussetzungen) direkt an den weiteren Server übermittelt, wobei die 5 Parameter bei Bedarf entsprechend modifiziert werden können.

Ergibt die Abfrage S24 hingegen, daß der Druckauftrag vom vorliegenden Server ausgeführt werden kann, wird in einer weiteren Abfrage S26 geprüft, ob der Druckauftrag direkt an die Druckeinrichtung vermittelt werden kann. Besitzt die Druckeinrichtung 10 eine entsprechende Netzankopplung, so kann ein direkter Link auf die Druckeinrichtung erzeugt werden, wodurch der Server nicht weiter durch den Druckauftrag belastet ist.

15 Ergibt die Abfrage aus dem Schritt S26, daß eine direkte Vermittlung des Druckauftrages an die Druckeinrichtung möglich ist, so wird diese im Schritt S27 ausgeführt.

Ist eine solche Vermittlung nicht möglich, so wird mit dem 20 Schritt S28 ein Link auf den Server bzw. sein Gateway gesetzt, um die Druckdaten im Schritt S29 über das Gateway zur Druckeinrichtung zu übermitteln. Nach Beendigung des Druckvorganges erfolgt eine Druckbestätigung S30 vom Server zum Client, womit das Verfahren beendet ist.

Bei diesem Verfahren wird der Druckauftrag automatisch zu einer geeigneten Druckeinrichtung weitergeleitet. Es wird sozusagen zwischen einzelnen Servern und Druckeinrichtungen ausgehandelt, wer zur Abarbeitung des Druckauftrages zur Verfügung steht und 30 geeignet ist. Man bezeichnet deshalb ein solches Verfahren auch als „Trading“.

Wie es oben dargestellt ist, ist das erfindungsgemäße Netzwerk insbesondere zur Ausbildung eines dezentralisierten Drucksystems geeignet, das einen oder mehrere Druckserver umfassen kann. Bei einer Internet-Anwendung des Druckservers kann jeder Internet-Client, der eine entsprechende Zugangsberechtigung zum 35

Server besitzt, diesen für seine Druckaufträge benutzen. Die technische Realisierung eines solchen dezentralisierten Drucksystems ist äußerst einfach und setzt lediglich die Installation eines erfindungsgemäßen Interpreters voraus, in dem die entsprechenden Befehle zum Ansteuern der Druckeinrichtungen bzw. der Vor- und Nachbearbeitungsgeräte eingerichtet sind. In einem einzigen Netzwerk können mehrere Server mit einem erfindungsgemäßen Interpreter versehen sein. Hierdurch ist es auch möglich, daß ein Druckauftrag zwischen mehreren Servern weitergereicht wird.

Die Erfindung ist jedoch nicht auf ein Drucksystem beschränkt, sondern kann zur Ansteuerung, Überwachung, Wartung, usw. von beliebigen Geräten verwendet werden. So bestehen zur Zeit erhebliche Bestrebungen Haushaltsgeräte netzwerkfähig zu machen. Mit einem erfindungsgemäßen Server können sie von einem Anwender über das Internet von einer beliebigen Stelle überprüft, abgefragt und evtl. in Betrieb gesetzt werden. Grundsätzlich ist die Verwaltung und Steuerung aller technischen Geräte mit dem erfindungsgemäßen Netzwerk möglich. Insbesondere eignet es sich zur Verwaltung von Datenverarbeitungssystemen, Telekommunikationssystemen und Vermittlungssystemen, wobei es insbesondere für überregionale Systeme von Vorteil ist, da der erfindungsgemäße Server von einer beliebigen Stelle des Netzwerkes aus angesteuert werden kann.

Ein weiterer, wesentlicher Vorteil des erfindungsgemäßen Netzwerkes ist, daß durch das Vorsehen eines Interpreters es nicht auf eine bestimmte Anwendung beschränkt ist, sondern durch die Gestaltungsmöglichkeit einer Computersprache es möglich ist, mit hochspracheähnlichen Sprachelementen die jeweilige Anwendung zu generieren. Hierdurch besitzt das erfindungsgemäße System eine maximale Flexibilität. Für die meisten Anwendungen gibt es bereits spezialisierte Programmteile, die lediglich in den Interpreter integriert werden müssen. Derartige Programmteile sind z.B. Gateways, Druckertreiber, Spooler oder sonstige singuläre Steuerprogramme. Insbesondere können spezialisierte

Kommunikationsprotokolle, wie SNMP oder DMI für die lokale -Datenübertragung zu den Druckeinrichtungen ausgenutzt werden, da diese Protokolle im Gegensatz zu dem HTTP-Protokoll einen wesentlich geringeren Protokolloverhead, geringere Responsezeiten und eine hohe Leistung besitzen und eine einfache Anwendung erlauben. Diese spezialisierten Protokolle bzw. spezifischen Schnittstellen können einem Anwender zugänglich gemacht werden, der sich weder mit dieser Technik auseinandersetzen muß, noch auf seinem Client eine entsprechende, in der Regel sehr aufwendige Software installieren muß, um eine Kommunikation zu denartigen spezialisierten Systemen herstellen zu können.

Der erfundungsgemäße Interpreter kann auf einen Datenträger gespeichert sein und von diesem oder über ein Netzwerk in einen Server geladen werden.

Die Erfindung kann folgendermaßen kurz zusammengefaßt werden:

Die vorliegende Erfindung betrifft ein Client-Server-Netzwerk, einen am Server des Netzwerkes installierbaren Interpreter und ein Verfahren zum Betreiben eines derartigen Client-Server-Netzwerkes.

Die Erfindung zeichnet sich dadurch aus, daß am Server Dateien abgelegt werden, die vom Client abgerufen werden können und die erfundungsgemäß sowohl am Client ausführbare Sprachelemente als auch am Server ausführbare Sprachelemente aufweisen. Am Server ist ein Interpreter vorgesehen, der die am Server ausführbaren Sprachelemente interpretiert und zur Ausführung bringt.

Nach einer bevorzugten Ausführungsform der Erfindung entsprechen die am Client ausführbaren Sprachelemente einer Markup-Sprache, wie z. B. SGML, XML, HTML, da dann der Benutzer beim Einrichten dieser Dateien seine bekannten Hilfsmittel zum Erstellen der Dateien, die in der Regel geläufige Textverarbeitungsprogramme sind, verwenden kann, um am Server des Netzwerkes individuelle Anwendungen vorzusehen, die von einem beliebi-

gen Client mit einem herkömmlichen Browser aufgerufen werden können.

Die Erfindung eignet sich besonders zur Steuerung von Geräten,  
5. insbesondere von Druckern und Drucksystemen und den entsprechenden Vor- und Nachbearbeitungsgeräten, da die Steuerungsin-  
telligenz zentral am Server hinterlegt wird und somit von vielen Clients benutzt werden kann, und der Datentransfer zwischen den Clients und dem Server gering gehalten wird.

10

Ein weiterer Aspekt der Erfindung ist, daß mit einfachen Mitteln ein Trading von z. B. Druckaufträgen zwischen mehreren Servern realisiert werden kann.

## Anhang

```

<head>
<TITLE>SNMP Response</TITLE>
</HEAD>

5   <BODY>

<!--
10  -----
      Kommentar: Programmanfang
  -----
//-->

15  <!--
  -----
      Kommentar: Funktion "STDefineString": wird von init aufgerufen
  -----
//-->

20  <PSUB FUNC="STDefineString">
    <PTAG FUNC="userDefineString" VALUE="SNMPVALUE%i%" VALUE="Offline"></PTAG>
    <PTAG FUNC="userDefineString" VALUE="REPLACEMENT%i%"></PTAG>
    <PTAG FUNC="userDefineString" VALUE="SnmpHLStart%i%"></PTAG>
25  <PTAG FUNC="userDefineString" VALUE="SnmpHLEnd%i%"></PTAG>
    <PTAG FUNC="userDefineString" VALUE="Stdout%i%"></PTAG>
    <PTAG FUNC="userDefineString" VALUE="Stdin%i%"></PTAG>
    <PTAG FUNC="userDefineString" VALUE="Stderr%i%"></PTAG>
</PSUB>

30  <!--
  -----
      Kommentar: Funktion "MTScanSystem"
  -----
//-->

35  <PSUB FUNC="MTScanSystem">
    <PTAG FUNC="userSNMPGateway"          // Kommentar: Abschnitt s2
      VALUE="SNMPVALUE%i%">
      VALUE="1"
      VALUE="160.120.17.%i%"
      VALUE="public"
      VALUE=".1.3.6.1.2.1.1.1.0">
40  </PTAG>

    <PBRANCH FUNC="userIf"              // Kommentar: Abschnitt s3
      VALUE="SNMPVALUE%i%">
      VALUE="FNC>WIN"
50    VALUE=<PTAG FUNC="userSNMPGateway" // Kommentar: Abschnitt s4
        VALUE="SNMPVALUE%i%">
        VALUE="1"
        VALUE="160.120.17.%i%">
        VALUE="public"
55    VALUE=".1.3.6.1.4.1.1552.92.2.0">
      </PTAG>
</PBRANCH>

60  <PBRANCH FUNC="userIf"          // Kommentar: Abschnitt s6
      VALUE="SNMPVALUE%i%">
      VALUE="FNC>SCO"
      VALUE=<PTAG FUNC="userSNMPGateway" // Kommentar: Abschnitt s7
        VALUE="SNMPVALUE%i%">
        VALUE="1"
65

```

```

      VALUE="160.120.17.%i"
      VALUE="public"
      VALUE=".1.3.6.1.3.1.1.1.1.0">
    </PTAG>
5   </PBRANCH>

    <PBRANCH FUNC="userIf"                                // Kommentar: Abschnitt s9
      VALUE="SNMPVALUE%i"
      VALUE="FNC>Offline"
10  VALUE=<PTAG FUNC="userSystem"
      VALUE="Stdout%i"
      VALUE="Stdin%i"
      VALUE="Stderr%i"
      VALUE="0"
15  VALUE="5000"
      VALUE="ping -n 1"
      VALUE="160.120.17.%i">
    </PTAG>
</PBRANCH>

20  <PBRANCH FUNC="userIf"                                // Kommentar: Abschnitt s10
      VALUE="Stdout%i"
      VALUE="FNC>Antwort"
      VALUE=<PTAG FUNC="userDefineString" // Kommentar: Abschnitt s11
        VALUE="SNMPVALUE%i"
        VALUE="PING"
      </PTAG>
</PBRANCH>

25  <PTAG FUNC="userCompose"                            // Kommentar: Abschnitt s12
      VALUE="SNMPVALUE%i"
      VALUE="false"

      VALUE="REPLACEMENT%i"
      VALUE="FNC>ISTREAM"
      VALUE="../images/imagestream.gif"

      VALUE="REPLACEMENT%i"
      VALUE="FNC>PRISMA"
40  VALUE="../images/prisma.gif"

      VALUE="REPLACEMENT%i"
      VALUE="FCS>PING"
      VALUE="../images/ping.gif"

      VALUE="REPLACEMENT%i"
      VALUE="FNC>Offline"
      VALUE="../images/offline.gif">
    </PTAG>
50  <PTAG FUNC="userCompose"                            // Kommentar: Abschnitt s14
      VALUE="SNMPVALUE%i"
      VALUE="false"

55  VALUE="SnmpHLStart%i"
      VALUE="FNC>ISTREAM"
      VALUE="<a HREF=\"http://www.ops.de\\">"

      VALUE="SnmpHLStart%i"
      VALUE="FNC>PRISMA"
      VALUE="<a HREF=\"http://160.120.17.%i/pjm.html\\" tar-
60  get=_blank\>"

      VALUE="SnmpHLStart%i"
      VALUE="FNC>Offline"
      VALUE="">
    </PTAG>

```

```

<PTAG FUNC="userCompose"                                // Kommentar: Abschnitt s15
  VALUE="SNMPVALUE%&i%" 
  VALUE="false"
5   VALUE="SnmpHLEnd%&i%" 
  VALUE="FNC>ISTREAM" 
  VALUE="</a>" 
10  VALUE="SnmpHLEnd%&i%" 
  VALUE="FNC>PRISMA" 
  VALUE="</a>" 
15  VALUE="SnmpHLEnd%&i%" 
  VALUE="FNC>Offline" 
  VALUE="">
</PTAG>
</PSUB>

20 <!--
----- Kommentar: Funktion "STPrintSystem: wird von s16 aufgerufen
-----//-->
25 <PSUB FUNC="STPrintSystem">
  <PTAG FUNC="userPreReplaceString" VALUE="SnmpHLStart%&i%"></PTAG>
  <img src=<PTAG FUNC="userPreReplaceString" VALUE="REPLACEMENT%&i%"></PTAG>
  width="80" height="80"
30  alt="160.120.17.%&i%" 
  <PTAG FUNC="userPreReplaceString" VALUE="SNMPVALUE%&i%"></PTAG>">
  <PTAG FUNC="userPreReplaceString" VALUE="SnmpHLEnd%&i%"></PTAG>
</PSUB>

35 <!--
----- Kommentar: ini1 - s1
-----//-->
40 <PTAG FUNC="userSetSNMPRetries" VALUE="2"></PTAG>
<PTAG FUNC="userSetSNMPTimeout" VALUE="2000"></PTAG>

<PTAG FUNC="userFor"
  VALUE="i"
  VALUE="200"
  VALUE="255"
  VALUE="1"
  VALUE="0"
  VALUE="STDefineString">
50 </PTAG>

<!--
----- Kommentar: Schritte s2-s15 → Aufruf der entsprechenden
----- Unterprogramme der Abschnitten s2-s15
-----//-->
55 <PTAG FUNC="userFor"
  VALUE="i"
  VALUE="200"
  VALUE="255"
  VALUE="1"
  VALUE="-1"
  VALUE="MTScanSystem">
60 </PTAG>
65 <PTAG FUNC="userFor"
  VALUE="i"
  VALUE="200"
  VALUE="255"
  VALUE="1"
  VALUE="-1"
  VALUE="MTScanSystem">
</PTAG>

```

```
<!--
-----
 Kommentar: Schritt s16
-----
5 //-->

<p>
<PTAG FUNC="userFor"
      VALUE="i"
10    VALUE="200"
      VALUE="255"
      VALUE="1"
      VALUE="0"
      VALUE="STPrintSystem">
15 </PTAG>
</p>

<!--
-----
20   Kommentar: Schritt s17 / Ende
-----
//-->

25 </BODY>
</HTML>
```

**Bezugszeichenliste**

5	1	erster Web-Server
	2	zweiter Web-Server
	3	erster Client
	4	zweiter Client
	5	Datenleitung (Intranet)
10	6	Datenleitung (Intranet)
	7	Datenverbindung (Internet)
	8	Gerät
	9	Gerät
	10	Serielle Leitung
15	11	SNMP-Verbindung
	12	Gateway
	13	Gerät
	14	Leitung
	15	Kommunikationsprogramm
20	16	WWW-Dienstprogramm
	17	Speicher
	17a	freier Speicherbereich
	17b	gesperrter Speicherbereich
	18	Interpreter
21		

**Verfahrensschritte**

- S1 Initialisierung
- 5 S2 Aufruf und Abfrage des SNMP-Gateway
- S3 Windows?
- S4 Windows-Drucksystem?
- S5 Abspeichern
- S6 UNIX?
- 10 S7 UNIX-Drucksystem?
- S8 Abspeichern
- S9 kein SNMP?
- S10 PING
- S11 PING?
- 15 S12 Abspeichern
- S13 UserCompose: Bild
- S14 UserCompose: Hyperlink Start
- S15 UserCompose: Hyperlink Ende
- S16 Einfügen des Hyperlinks in HTML-Seite
- 20 S17 Ende
- S18 Eingabe des Druckauftrages
- S19 Übertragung vom Client zum Server
- S20 Aktualisierung einer Datenbasis
- S21 Auswertung der Datenbasis
- 25 S22 Geeignete Druckeinrichtung?
- S23 Nachricht an Client
- S24 Kann Server Druckauftrag ausführen?
- S25 Vermittlung an weiteren Server
- S26 Kann an Druckeinrichtung vermittelt werden?
- 30 S27 Vermittlung an Druckeinrichtung mit direktem Link
- S28 Link auf Server
- S29 Übermittlung über Gateway
- S30 Druckbestätigung

**Ansprüche**

5    1. Netzwerk vom Client-Server-Typ, das zumindest einen Client (3, 4) mit zumindest einem Server (1, 2) verbindet, wobei am Server (1, 2) gespeicherte Dateien vom Server (1, 2) zum Client (3, 4) übertragen werden, wenn sie der Client (3, 4) durch Senden einer entsprechenden Dateiadresse an den Server (1, 2) abruft, und die Dateien sowohl am Client (3, 4) ausführbare Sprachelemente als auch am Server (1, 2) ausführbare Sprachelemente enthalten, und am Server (1, 2) ein Interpreter zum Interpretieren und Ausführen der am Server ausführbaren Sprachelemente vorhanden ist.

10    2. Netzwerk nach Anspruch 1,  
dadurch gekennzeichnet,  
daß der Interpreter (18) am Server (1, 2) derart ausgebildet ist, daß die am Server ausführbaren Sprachelemente nach dem Aufrufen der Dateien durch einen Client und vor der Übermittlung der Dateien zum Client (3, 4) am Server (1, 2) ausgeführt werden.

15    3. Netzwerk nach Anspruch 1 oder Anspruch 2,  
dadurch gekennzeichnet,  
daß die Dateiadresse dem URL-Format entspricht und der Server (1, 2) ein Web-Server ist, so daß die Dateien mit einem am Client (3, 4) installierten Internet-Browser abgerufen werden können.

20    4. Netzwerk nach einem der Ansprüche 1 bis 3,  
dadurch gekennzeichnet,  
daß die am Server (1, 2) gespeicherten und vom Client (3, 4) abrufbaren Dateien dem Format einer Markup-Sprache

(SGML, XML, HTML) entsprechen, das durch die am Server ausführbaren Sprachelemente erweitert ist.

5. Netzwerk nach einem der Ansprüche 1 bis 4,

**dadurch gekennzeichnet,**

daß am Server (1, 2) zumindest ein Gateway (12) installiert ist, das eine Datenverbindung zu einem weiteren logischen und/oder physikalischen System (Gerät, Netzwerk, Queue, Computer) herstellen kann, wobei die Daten des weiteren Systems ein anderes Format als die zwischen dem Server (1, 2) und dem Client (3, 4) ausgetauschten Daten aufweisen, und das Gateway (12) sowohl die einkommenden als auch die ausgehenden Daten in die entsprechenden Datenformate automatisch umsetzt.

15

6. Netzwerk nach Anspruch 5,

**dadurch gekennzeichnet,**

daß jeweils ein Gateway (12) zum Umsetzen der Daten in eines oder mehrere der folgenden Formate vorgesehen ist:

20

SNMP, LP, PJMweb, ftp.

7. Netzwerk nach Anspruch 5 oder 6,

**dadurch gekennzeichnet,**

daß das bzw. die Gateways (12) im Interpreter integriert sind und durch Sprachelementen des Interpreters (18) aufgerufen werden können.

8. Netzwerk nach einem der Ansprüche 1 bis 7,

**dadurch gekennzeichnet,**

30

daß am Server (1, 2) Programme zum Ansteuern zumindest eines Druckers und/oder Vor- oder Nachverarbeitungsgeräte, wie z.B. eines Druckertreibers oder eines Spoolers installiert sind, und diese Programme vom Interpreter (18) aufgerufen werden können.

35

9. Interpreter für ein Netzwerk, insbesondere nach einem der vorhergehenden Ansprüche, der an einem Server (1, 2) eines

Client-Server-Netzwerkes installierbar ist und zum Interpretieren und Ausführen von am Server (1, 2) ausführbaren Sprachelementen ausgebildet ist, die in am Server (1, 2) abgespeicherten Dateien enthalten sind, wobei diese Dateien von einem Client (3, 4) mittels der Übertragung einer Adresse abgerufen werden können und zusätzliche am Client (3, 4) ausführbare Sprachelemente enthalten.

- 5 10. Interpreter nach Anspruch 9,  
dadurch gekennzeichnet,  
daß der Interpreter einen Befehl (userCompose) zum Erzeu-  
gen von String-Einträgen in der Datei aufweist.
- 15 11. Interpreter nach Anspruch 9 oder 10,  
gekennzeichnet durch  
einen Befehl (userPreReplaceString, userPostReplaceString)  
zum Setzen von String-Einträgen an eine vorbestimmte Stel-  
le der Datei.
- 20 12. Interpreter nach einem der Ansprüche 9 bis 11,  
gekennzeichnet durch  
einen Befehl zum Einlesen eines vom Client (3, 4) an den  
Server (1, 2) übertragenen Strings und zum Abspeichers des  
Strings in eine vorbestimmte Variable (userGetCGIString).
- 25 13. Interpreter nach einem der Ansprüche 9 bis 12,  
gekennzeichnet durch  
einen Befehl zum Aufrufen eines Gateways und Abfragen ei-  
nes mit dem Gateway verbundenen Systems.
- 30 14. Interpreter nach einem der Ansprüche 9 bis 13,  
dadurch gekennzeichnet,  
daß der Interpreter eine Gruppe von Client-Befehlen um-  
faßt, die sowohl vom Client aus als auch vom Server aus  
35 aufgerufen werden können, und eine Gruppe von Serverbefeh-  
len umfaßt, die nur vom Server aus aufgerufen werden kön-  
nen.

15. Interpreter nach einem der Ansprüche 9 bis 14,  
**dadurch gekennzeichnet,**  
daß der Interpreter auf einen Datenträger gespeichert ist.  
5
16. Verfahren zum Betreiben eines Netzwerkes nach einem der  
Ansprüche 1 bis 8, wobei  
das Netzwerk ein Client-Server-Netzwerk ist, und am Server  
(1, 2) gespeicherte Dateien vom Server (1, 2) zum Client  
10 (3, 4) übertragen werden, wenn sie der Client (3, 4) durch  
Senden einer entsprechenden Dateiadresse an den Server (1,  
2) abruft, und die Server den Clients jeweils einen oder  
mehrere Dienste anbieten, wobei  
bei einer Client-Anfrage nach einem bestimmten Dienst mit  
bestimmten, den Dienst zugrundeliegenden Parametern, der  
abgefragte Server bestimmt, ob er den Dienst erfüllen  
kann, und wenn der Server feststellt, daß er den Dienst  
nicht erfüllen kann, er einen weiteren Server oder an ein  
am Netzwerk angeschlossenes Gerät an den Client vermit-  
telt, das den Dienst ausführen kann.  
15
17. Verfahren zum Betreiben eines Netzwerkes nach Anspruch 16,  
**dadurch gekennzeichnet,**  
daß einer der von den Servern angebotenen Dienste die Aus-  
führung eines Druckauftrages ist, und  
der Server den Druckauftrag an einen anderen Server oder  
direkt an eine Druckeinrichtung weiterleitet, wenn der  
Server den Druckauftrag nicht selbst ausführen kann.  
20
18. Verfahren zum Betreiben eines Netzwerkes nach Anspruch 16  
oder 17,  
**dadurch gekennzeichnet,**  
daß der Server eine Datenbank aufweist, in der Informatio-  
nen zu den im Netzwerk angebotenen Diensten abgespeichert  
30 sind, so daß bei einer Client-Anfrage anhand dieser Daten-  
banken ermittelt werden kann, ob der gewünschte Dienst im  
Netzwerk vorhanden ist.  
35

19. Verfahren zum Betreiben eines Netzwerkes nach einem der Ansprüche 16 bis 18,  
**dadurch gekennzeichnet,**  
5 daß die Vermittlung an einen weiteren Server oder ein an das Netzwerk angeschlossenes Gerät durch Erzeugen der Adresse des weiteren Servers oder des Gerätes und durch Übermittlung der Adresse an den anfragenden Client ausgeführt wird.

10 20. Verfahren zum Betreiben eines Netzwerkes nach einem der Ansprüche 16 bis 19,  
**gekennzeichnet durch**  
einen Interpreter nach einem der Ansprüche 9 bis 15.

15

## Zusammenfassung

Die vorliegende Erfindung betrifft ein Client-Server-Netzwerk, einen am Server des Netzwerkes installierbaren Interpreter und  
5 ein Verfahren zum Betreiben eines derartigen Client-Server-Netzwerkes.

Die Erfindung zeichnet sich dadurch aus, daß am Server Dateien abgelegt werden, die vom Client abgerufen werden können und die  
10 erfindungsgemäß sowohl am Client ausführbare Sprachelemente als auch am Server ausführbare Sprachelemente aufweisen. Am Server ist ein Interpreter vorgesehen, der die am Server ausführbaren Sprachelemente interpretiert und zur Ausführung bringt.

15 Nach einer bevorzugten Ausführungsform der Erfindung entsprechen die am Client ausführbaren Sprachelemente einer Markup-Sprache, wie z. B. SGML, XML, HTML, da dann der Benutzer beim Einrichten dieser Dateien seine bekannten Hilfsmittel zum Erstellen der Dateien, die in der Regel geläufige Textverarbeitungsprogramme sind, verwenden kann, um am Server des Netzwerkes individuelle Anwendungen vorzusehen, die von einem beliebigen Client mit einem herkömmlichen Browser aufgerufen werden können.

20 Die Erfindung eignet sich besonders zur Steuerung von Geräten, insbesondere von Druckern und Drucksystemen und den entsprechenden Vor- und Nachbearbeitungsgeräten, da die Steuerungsin-telligenz zentral am Server hinterlegt wird und somit von vielen Clients benutzt werden kann, und der Datentransfer zwischen  
30 den Clients und dem Server gering gehalten wird.

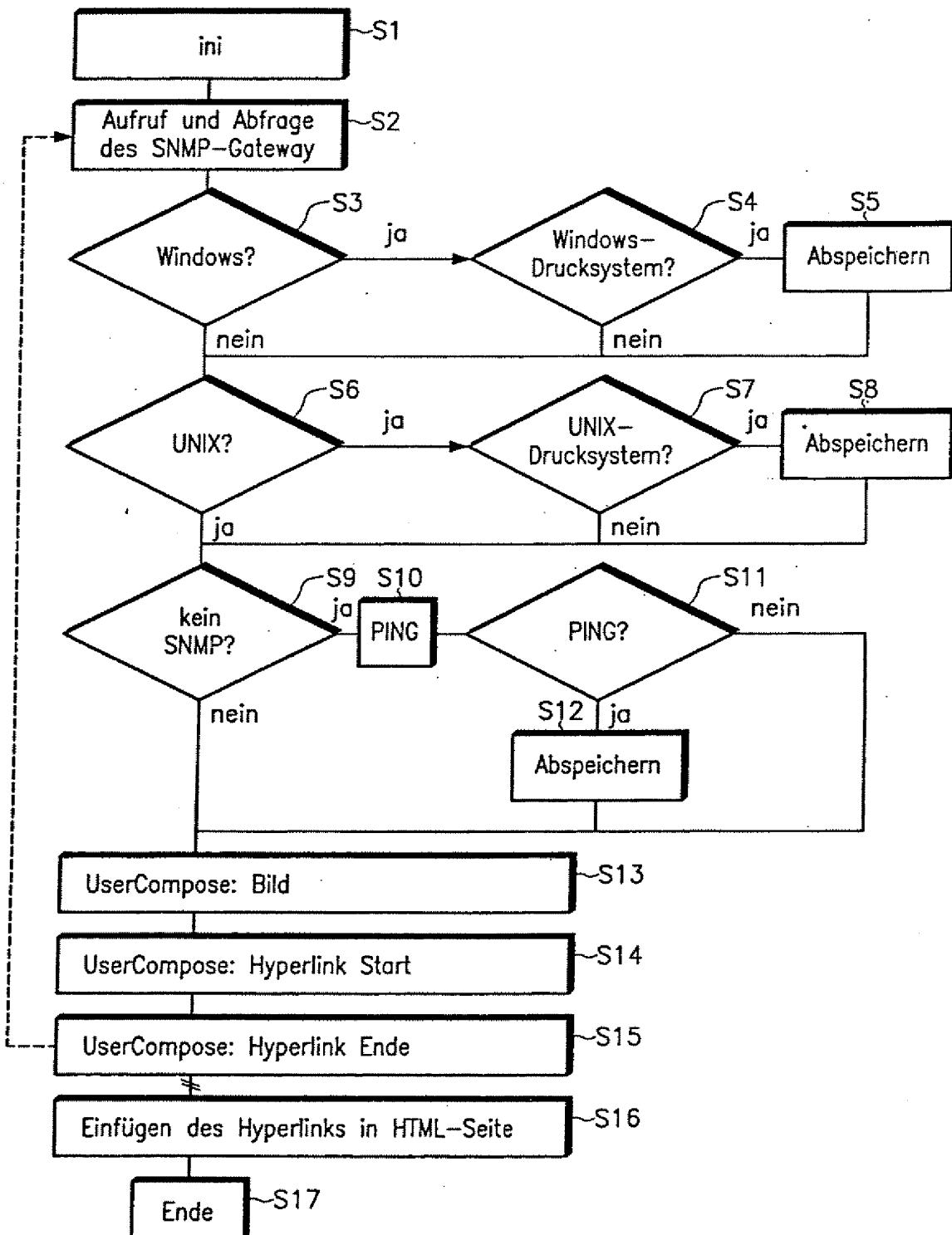
Ein weiterer Aspekt der Erfindung ist, daß mit einfachen Mitteln ein Trading von z. B. Druckaufträgen zwischen mehreren Servern realisiert werden kann.

35

(Figur 2)

990503 DE

# Z U S A M M E N F A S S U N G



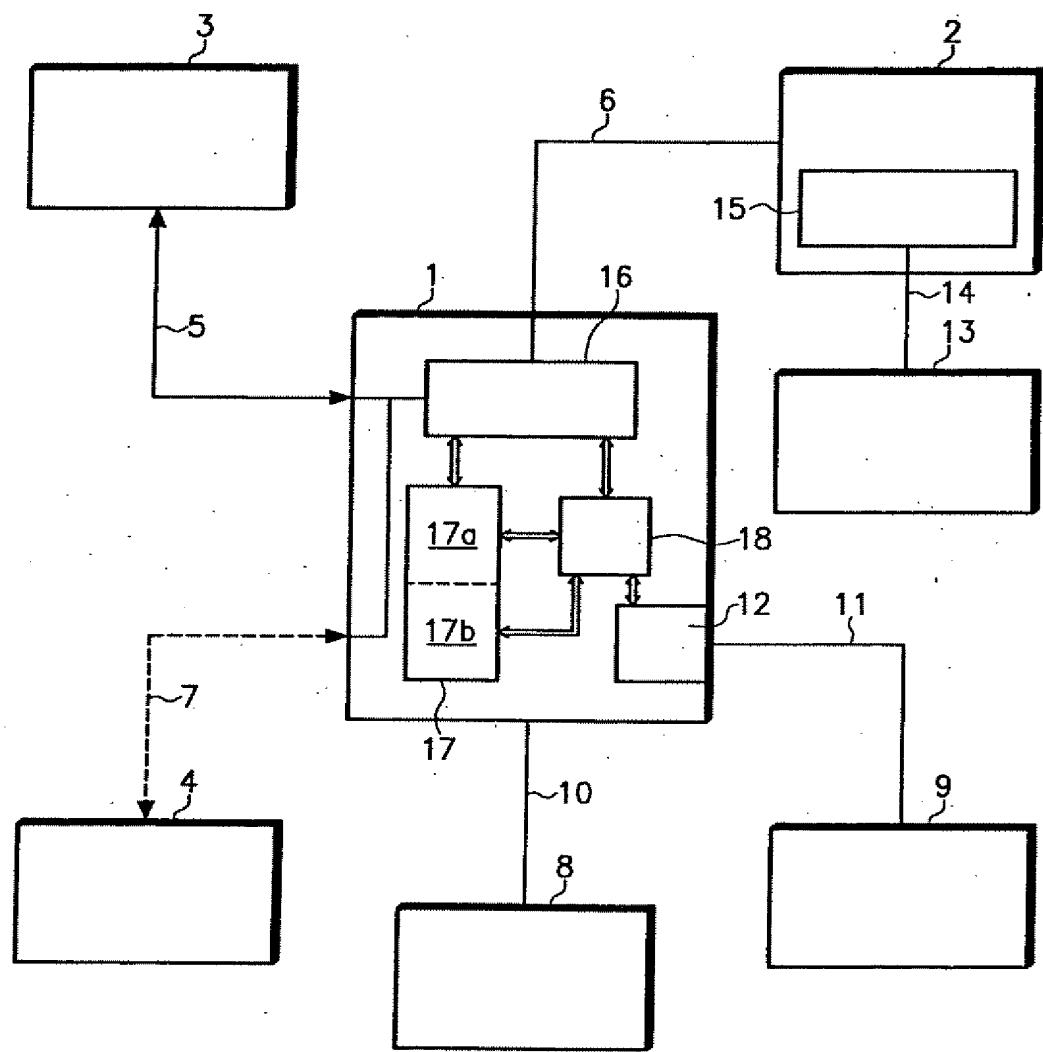


FIG. 1

2/3

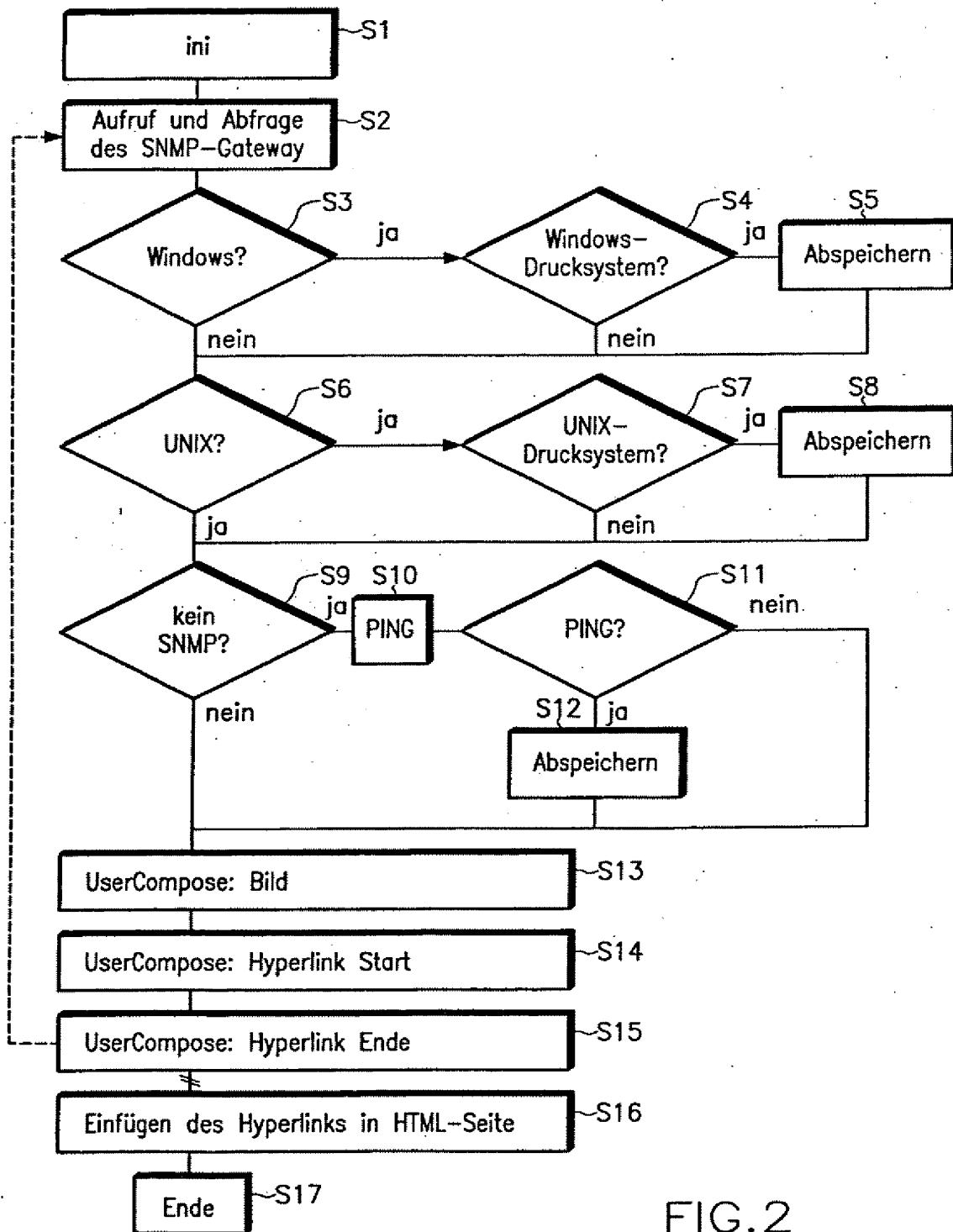


FIG.2

3/3

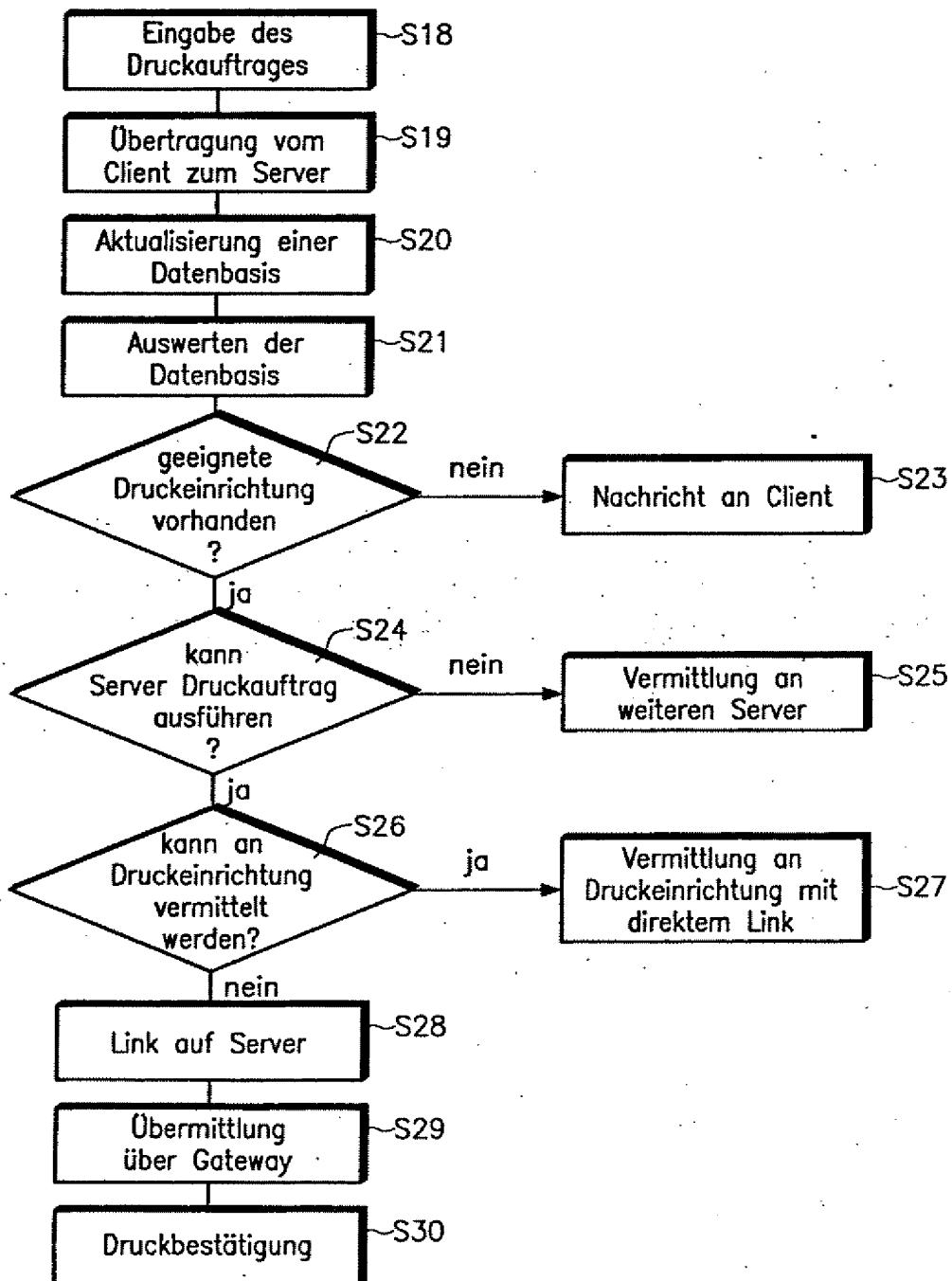


FIG.3